2

WL-TR-91-4012

AN EXPERT SYSTEM FOR THE IDENTIFICATION OF SILICON IMPURITIES USING FTIR SPECTROSCOPY

Ronald E. Perrin
Steven B. Fairchild
Electronic and Optical Materials Branch
Electromagnetic Materials & Survivability Division

Final Report for Period Jan 89 to Jun 90

APRIL 1991

Approved for Public Release; Distribution Unlimited.

MATERIALS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

RONALD E. PERRIN
Electronic & Optical Materials Branch

ROBERT L. DENISON, Chief
Electronic & Optical Materials Branch
Electromagnetic Mat'ls & Surv. Division

STEVEN B. FAIRCHILD
Hardened Materials Branch

FOR THE COMMANDER

William R Woody
WILLIAM R. WOODY, Chief
Electromagnetic Mat'ls & Surv. Division
Materials Directorate

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/MLPO. WPAFB, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY 2b DECLASSIFICATION / DOWNGRADING SCHEDULE	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for Public Release; distribution unlimited.
4 PERFORMING ORGANIZATION REPORT NUMBER(S) WL-TR-91-4012	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Materials Directorate Wright Laboratory 6b OFFICE SYMBOL *(If applicable)* WL/MLPO	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS *(City, State, and ZIP Code)* Wright-Patterson AFB, OH 45433-6533	7b ADDRESS *(City, State, and ZIP Code)*
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS *(City, State, and ZIP Code)*	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
61102F	2306	Q1	06

11 TITLE *(Include Security Classification)*

An Expert System for the Identification of Silicon Impurities Using FTIR Spectroscopy

12 PERSONAL AUTHOR(S)

Ronald E. Perrin and Steven B. Fairchild

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT *(Year, Month, Day)*	15 PAGE COUNT
Final	FROM 1/89 TO 6/90	91 Apr 30	121

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
14	02	
20	06	

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

Expert-System, electronic materials, impurities, Fourier Transform Infra-Red Spectroscopy

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

An expert-system for the analysis of the infra-red spectra of silicon based electronic materials has been developed. The Fourier Transform Infra-Red Spectroscopy system produces characteristic spectral lines for impurities present in silicon materials under analysis. A MicroVax workstation with a commercially available expert-system shell is attached to the spectrometer. By examining these spectral line patterns the expert-system is able to deduce the identities and concentrations of the unknown impurities. The use of this program greatly reduces the analysis time required for each sample and produces more consistent results.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Ronald Perrin or Steve Fairchild	22b TELEPHONE *(Include Area Code)* 513/255-4474	22c OFFICE SYMBOL WL/MLPO

Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(continued)

LIST OF TABLES

1.0 INTRODUCTION

The Wright Laboratory, Electromagnetic Materials Division (WL/MLP) conducts research to develop electronic materials of future interest to the Air Force. These materials can include high purity silicon and gallium arsenide. These materials are used as infra-red detectors in space based applications as well as being used as a basis material for high-speed integrated circuits in the next generation of computers. The impurities and defects in these materials determine the operational limitations of the devices constructed from them. One of the optical characterization techniques used to detect these defects is low temperature Fourier Transform Infra-Red spectroscopy (FTIR). The material being examined is cooled to a cryogenic temperature and then excited by an infra-red light source. The absorption and transmittance of infra-red radiation by these impurities and defects produces a unique spectrum. By examining the spectrum, an experienced spectroscopist can determine the presence of each impurity as well as its concentration in the material[1]. The locations of peaks, the relative peak heights, and the shape of the background in the resulting spectral plot are used to characterize the material's properties. The presence of unwanted impurities in these materials degrades the performance of the end application devices.

1.1 BACKGROUND

Modern research and development laboratories can benefit from applying new automation techniques. The application of these new technologies can result in more accurate, more consistent, and faster data acquisition, thus maximizing personnel resources. WL/MLP decided that an Expert System application to Fourier Transform Infra-Red Spectroscopy (FTIR) would be an optimum project to pursue in terms of AI applications in the modern laboratory. In 1986 and 1987, two separate groups of artificial intelligence (AI) experts from universities and industry[2] surveyed the research being performed at WL/MLP). All current and future areas of research being performed within the WUD 48 and WUD 50 in-house laboratories were examined. The groups were asked to identify laboratory systems which could benefit from the application of artificial intelligence technology and recommend useful hardware and software products. After interviewing the researchers and being briefed on the details of each laboratory's data acquisition and analysis equipment, the AI working groups generated final reports. These reports outlined which research areas would benefit from applying AI technology and produce the greatest pay-off to the Air Force. This pay-off would be realized in terms of higher productivity and more accurate data analysis if automated using AI technology. The results of one survey are listed in Table 1. Note that the areas referenced as FTIR-2 and FTIR-3 show the highest ranking in terms of feasibility and pay-off. We proposed that an expert system that

Table 1. Results of AI feasibility survey.

OVERALL CONCLUSIONS AND RECOMMENDATIONS

The rankings given in the table for the various AI application opportunities are based on technical considerations alone. (1 = highest potential)

Opportunity Number and Name		Recommendation	Rank
MBE-1	Utilize Past Experience	pursue/guarded/ build data-base first	3
MBE-2	Interp. RHEED Oscill.	Do not pursue re-examine later	
MBE-3	Interp. Optical Probes	Do not pursue re-examine later	
MBE-4	Qual. Simul. of Growth	Pursue/guarded/ build data-base first	4
PVD-1	Correction Analysis	Do not pursue	
PVD-2	Detection Analysis	Do not pursue	
PVD-3	Intell. Front-End to Chem. Abstracts Data-Base	Pursue with constraints	2
FTIR-1	Initial Set of Experiments	Do not pursue	
FTIR-2	ID Known Centers and Locate Unknown Centers	Pursue	1
FTIR-3	Categorize and Assess Significant Unknown Centers	Pursue	2
FTIR-4	Correlate Unknown Centers	Pursue/guarded build data-base first	3
FTIR-5	Formulate Strategy for Further Experiments	Pursue/guarded/ build data-base first	4

could address the areas described in Table 1 as FTIR-2 and 3 be designed. This project would not only benefit the FTIR laboratory but lessons learned would be extremely valuable for future AI projects within WL/MLP.

1.2 PROBLEM STATEMENT

The task of analyzing FTIR spectra is complex and tedious. Simply matching peaks with known impurity wavenumber locations tells the researcher very little. The most common impurities, aluminum, boron, gallium, indium, carbon and oxygen, all occur in the same area of the spectrum (0-2000 wavenumbers). The presence of a singularity, or peak, in the spectrum at a certain wavenumber could be the result of dust present in the sample chamber or it could indicate the presence of an impurity or sample defect. The thickness of the sample as well as the method of mounting it also affects the shape of the spectrum. Sixty-hertz pickup and its harmonics show up in spectra as well as noise spikes that occasionally occur due to power line transients. The researcher must carefully look for peak patterns knowing from experience that many false conclusions could be reached if certain rules of thumb are not followed. It is this well defined set of rules and heuristics which, if incorporated into a knowledge base, could allow a less experienced researcher or aide to accurately characterize samples. This type of knowledge base could be useful for basic research as well as for routine material quality control testing.

Since the area of a selected peak is used to calculate its impurity concentration, the expert system must be capable of defining the baseline of a peak. While this task may seem simple, it must be remembered that peaks can be very weak if the concentrations are small. Peaks can also occur in very noisy areas of the spectrum or may be present where the spectral background is increasing or decreasing. Choosing the baseline of a peak in a noisy area is a subjective choice based on the past experience of the researcher. Often two expert spectroscopists will calculate peak areas differing by 30% or more when examining the same peak in a noisy background. The process of calculating peak areas accurately can be extremely time consuming. Researchers have, in the past, resorted to cutting the peak of interest out of a spectral plot and weighing it on a set of sensitive scales. This weight was compared to the weight of a unit area from the same type of graph paper. Obviously an easier, more consistent method is needed. The application of the proposed expert-system would generate more consistent analyses and make a tedious and exacting task quick and routine.

1.3 PROJECT OBJECTIVES

One of the most difficult tasks in designing this project was deciding what the expert system should actually be able to do. There is always a trade-off between what is needed by the end user and what is feasible to design given the constraints of of the expert-system shell, the skills of the designer, and a reasonable completion date. There is a saying, "Anything can be made to work if you tinker with it long enough." We wanted to limit the scope of this project to be realistic given our expert system design experience, time constraints, and our other job responsibilities.

It was decided that the expert system, given an FTIR absorption coefficient spectrum, should be able to identify all impurities present, their concentrations in the material, and all the associated impurity lines present. If the spectrum is a photoconductivity spectrum, it should be able to identify all impurities present and their associated line numbers.

To accomplish this, a knowledge base had to be designed that incorporated all the necessary information about how to identify these impurities in a silicon spectrum. The theoretical background for this knowledge base came from two sources. The first source is the knowledge obtained from extensive question and answer sessions with experienced spectroscopists in this particular area of materials analysis. This part is typically refered to as 'knowledge engineering.' The second source of information is technical papers and reports directed in this area of research. Together these sources provide a realistic starting point for constructing a useful expert system. Table 2 is taken from such a paper[2] and lists all the lines (i.e., peaks) by wavenumber that can occur in a silicon spectrum owing to the impurities of interest.

There should be minimal user interaction. The user could be asked if they agree with interim results; if not, the user has the authority to modify these, but the final conclusions are determined by the expert system. A graph of the spectrum being examined should be

Table 2. All of the peaks that can occur in a silicon spectrum due to the presence of the impurities.

Line	BORON	ALUMINUM	GALLIUM	INDIUM	THALLIUM
1	244.96	442.73	469.75	1145.41	1876.1
2	278.19	471.93	------	1175.82	1906.7
3	309.32	------	541.45	1208.00	1937.4
4	319.32	516.52	548.23	1216.31	1947.9
4a	319.97	524.11	550.57	1218.57	
4b	321.9	525.56	552.02	1219.46	
5	334.49	535.16	564.2	1232.2	
6a	338.03	------	-----	-----	
6b	339.24	----	567.82	----	
6	340.05	538.71	568.71	1236.56	1968.0
7	342.07	541.21	570.97	1239.05	
8	344.73	543.87	573.72	1241.96	
8a	346.18	545.56	575.49	1243.33	
9a	348.12	546.94	576.94	-----	
9	348.76	547.74	577.67	1245.83	
10a	350.78	549.76	579.72	1247.93	
10b	351.99	550.24	------	------	
10	352.55	557.53	581.30	1249.62	
10c	353.12	552.74	581.78	-----	
10d	354.49	553.47	583.23	1250.83	
11a	----	554.92	584.28	----	
11b	356.5	555.49	584.93	1253.41	
11	357.07	556.21	585.73	1254.3	
Ei	358.04	556.78	586.62	1254.79	1985.8
2p'	668.57	867.31	897.23	1565.4	2285.0
3p'	692.85	891.75	921.51	1589.84	2309.2
4p'	701.32	900.22	929.98	1598.47	
5p'	-----	-----	934.01	-----	
Ei*	712.93	911.75	941.5	1609.76	
X1	184	338	347	799	
X2	220	372	381	831	
X3	250	404	----	863	
X4	261	414	423	873	
Ei(x)	299	454	460	909	

displayed with all peaks marked by the expert system. If any peaks of interest have been missed, the user should be able to add these with the mouse and cursor. The user should not be allowed to delete peaks that the system has marked; it is up to the expert system to conclude whether marked peaks are significant. All impurities identified must be present in the patterns that experienced spectroscopists recognize as being necessary to insure a high degree of confidence. Noise and spurious peaks present in the spectrum should be ignored. Indeterminate peaks should be noted and classified as unknown. Finally, a report containing this information should be prepared in a concise format and be displayed and printed upon conclusion of the analysis.

2.0 TUTORIAL

The purpose of an AI knowledge based system is to transform the line of thought of a human expert into a set of formalized symbolic operations so as to reproduce his or her expertise.[3] Knowledge design is the process of transferring this formulated knowledge from humans to the machine. Once the computer has this set of rules formulated from human logic, it can perform deductive or inductive reasoning and arrive at the same conclusion as the human expert when given the appropriate data.

The methodology involved in knowledge engineering involves two closely related critical dimensions: the *reasoning dimension* and the *representation dimension*.[3] The reasoning dimension is composed of the *rules* that constitute the logic the expert system uses to perform reasoning. The representation dimension is composed of *objects* and *classes*, which can be thought of as symbolic representations of the data that the rules reason upon. Therefore, objects and classes are complimentary to rules.

The reasoning dimension and the representation dimension are closely interrelated to provide an efficient network. They allow ease in knowledge integration, which is the reduction of thought mechanisms to more understandable concepts. Once knowledge integration has been accomplished, powerful symbolic computations can be performed which allow automation of the problem solving process.[3]

2.1 THE REASONING DIMENSION

To represent reasoning, an expert system uses rules. Rules are symbolic structures which express deductive or evocative progressions in a reasoning path. By deductive we mean that a rule can be used to verify conditions in another rule. By evocative, we mean that a rule can trigger the activation or the evaluation of other rules.[3]

A rule is a chunk of knowledge representing a situation and its immediate consequences. The format of a rule is a symbolic structure of the type

if ... then ... and do...

where ***if*** is followed by a set of *conditions*, ***then*** by a *hypothesis* or goal which becomes TRUE when the conditions are met, and ***do*** by a set of *actions* to be undertaken as a result of a positive evaluation of the conditions of the rule.[3]

2.2 THE REPRESENTATION DIMENSION

The representation dimension is composed of a set of interrelated objects which represent the data or events that the rules act upon. Since the rules reason on a representation of the problem, the reasoning and representation dimension are inherently related.[3]

The representation of knowledge is accomplished with three symbolic tools: *objects*, *properties*, and *classes*. Various combinations of these three components provide an efficient method of representating the problem upon which the rules act.[3]

An object is an elementary unit of description. Everything is an object. A property is a characteristic of an object. A class is a collection of objects that share properties. As illustrated in Figure 1, objects are illustrated by triangles, properties by squares, and classes by circles.[3]

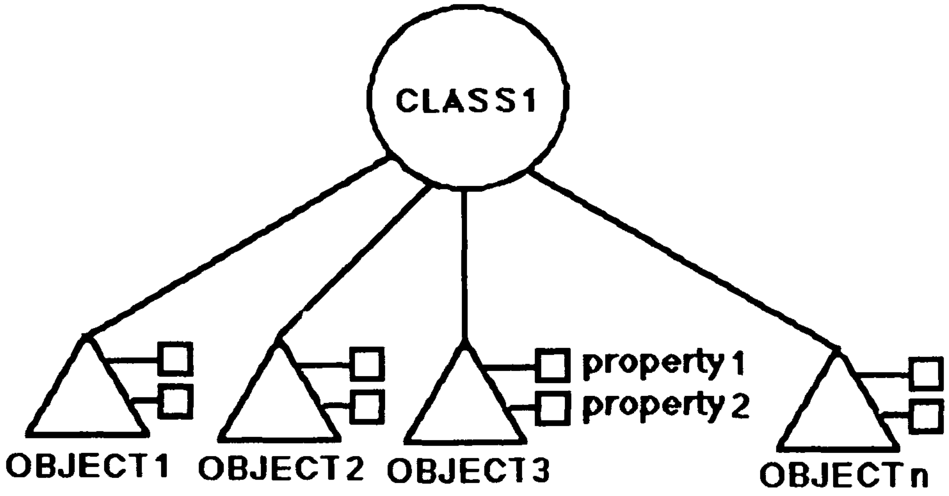


Figure 1. The symbolic representation of a class with its related objects and properties.[3]

2.3 EXAMPLE

This relationship between classes, objects, and properties and their connection to rules can be seen in the following example, which introduces the concept of knowledge base design to spectroscopic analysis.

A spectroscopist can examine the optical spectrum of a material to determine its composition. For example, pure silicon can always be identified by its specific characteristic spectrum. If a certain spectrum of silicon contains peaks that would not show up in a spectrum of pure silicon, then there are impurities present in the silicon sample from which the spectrum was generated. From previous experimental knowledge we know that various impurities identify themselves by the specific wavenumbers at which these peaks appear.

For example, if the impurity indium is present in a silicon sample, it can manifest itself by causing as many as 20 additional peaks to appear in the spectrum, three of which can be used to confirm the presence of indium. These peaks will be referred to as Indium_Lines. The confirming peaks are Indium_Line_1, Indium_Line_2, and Indium_Line_4. All three of these lines must be present for indium to exist. Figure 2 shows an example of a peak pattern in a silicon spectrum that identifies the presence of indium.

Figure 3 illustrates the object structure. Indium_Lines is a class, and Indium_Line_1, 2 & 4 are the objects. As previously stated, a class is a group of objects that have the same properties. Since each object Indium_Line_*n* will be identified by a specific wavenumber, each object will be given the property WN, for wavenumber. Each object will also be given the property EXIST, which will be used to identify whether or not that particular peak is present.

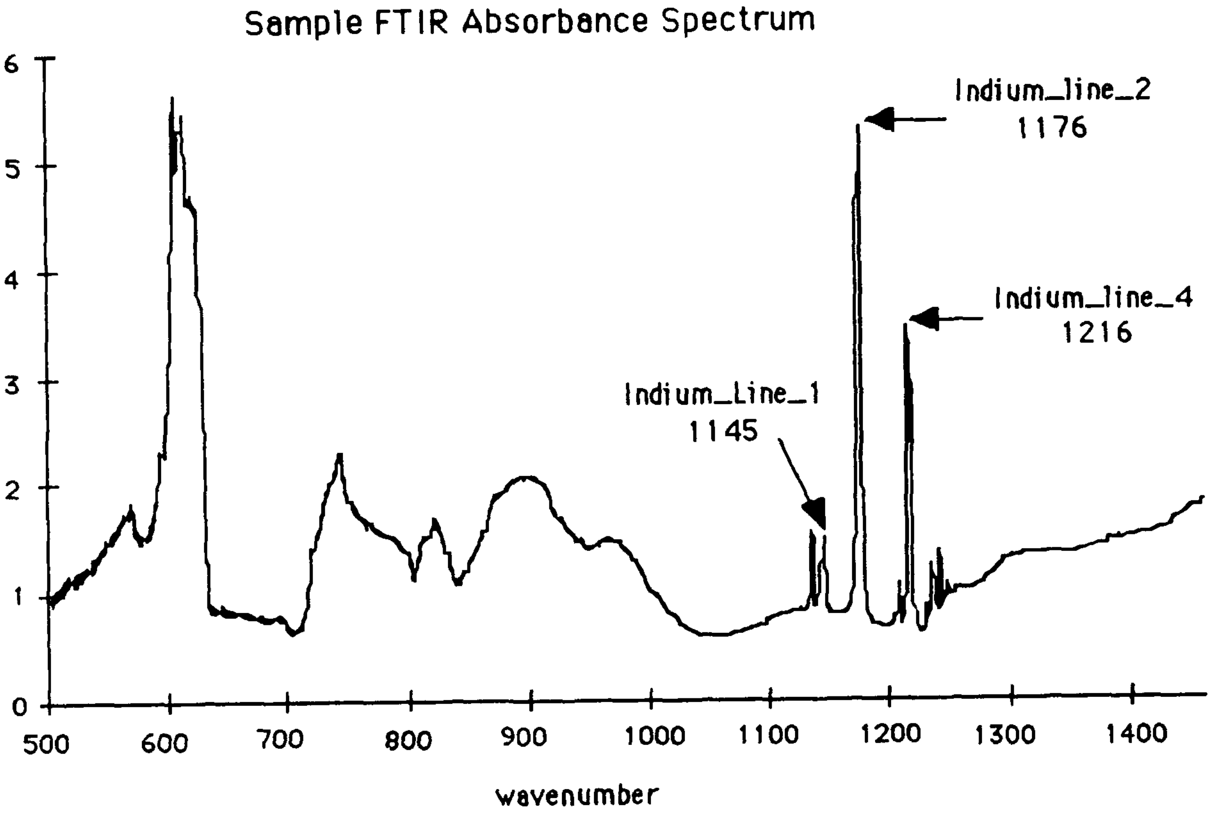


Figure 2. A silicon spectrum showing evidence of the presence of indium.

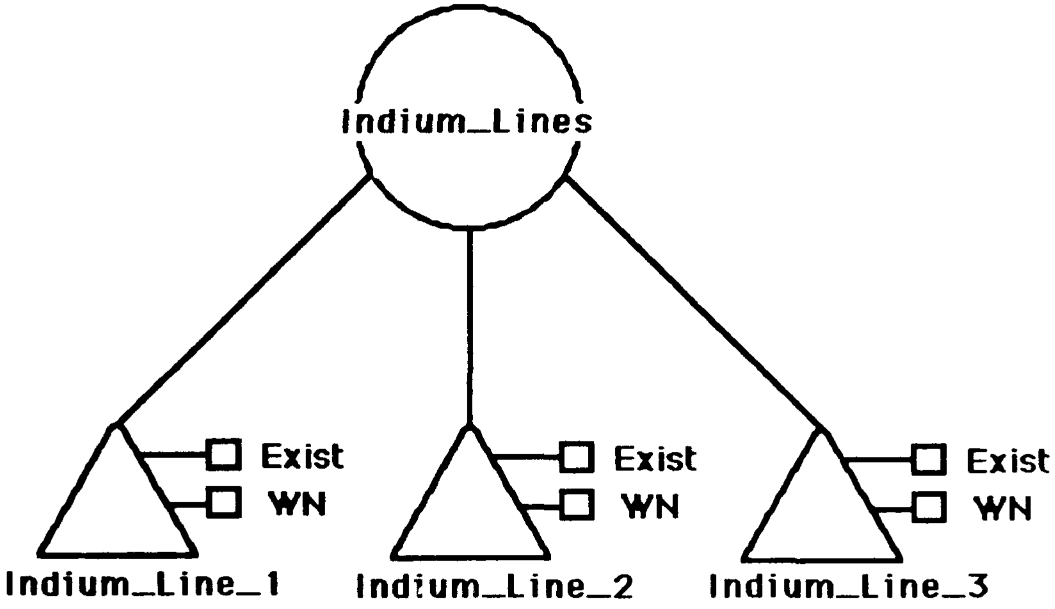


Figure 3. Symbolic representation of the class Indium_Lines.

Rules must operate on data, and data in the object network is represented by *slots*.[3] A slot is the value that is held in the property of an object:

object.property* = *slot

For this object network, a slot would be Indium_Line_1.EXIST and this slot (datum) would have a boolean value. Another slot would be Indium_line_4.WN, and it would have a numeric value. For example, the wavenumber for the third indium line is 1216, so the value held in the slot Indium_line_4.WN is 1216.

Rules operate on the objects, which are a symbolic representation of the problem.[3] A rule can be designed that will confirm whether or not indium is present in the silicon spectrum. This rule attempts to confirm the presence of the confirmation pattern.

if Indium_Line_1.EXIST is *true*
and Indium_Line_2.EXIST is *true*
and Indium_Line_4.EXIST is *true*
then Indium_Present is *confirmed*

In this rule, there are three *conditions* which lead to the evaluation of a *hypothesis*. The conditions are the tests for the presence of the three indium lines, and the hypothesis is the attempt to confirm the presence of indium. All three conditions must be true in order for the hypothesis to be confirmed, i.e., all three indium lines must be present in order for the presence of indium to be confirmed.

This rule operates on the slots that were set up in the object network. It will evaluate the slots Indium_line_n. EXIST. (Assume that a previous rule has determined the values for Indium_line_n.WN.) The values of these slots will determine whether or not the hypothesis is confirmed. The name of the hypothesis is Indium_ Present, which has a Boolean value. If all the data values (i.e., Indium_line_*n*.EXIST) that occur in the *conditions* evaluate to true, then the hypothesis Indium_Present will evaluate to true. If any of the datum evaluate to false, the hypothesis Indium_ Present will evaluate to false.

Therefore, if the pattern of the three indium lines is found, this rule will confirm that indium is indeed present. If any one of the three lines does not exist, then the rule confirms that indium is not present, since the whole pattern must be present to confirm the presence of indium.

This example has introduced the concept of knowledge base design. The most important part of knowledge base design is the inherent relationship between rules and the object network. As illustrated, the object network is composed of classes, objects, and properties, which are a symbolic representation of the problem. Rules can be related to the object network through their ability to evaluate data. Data are represented by a slot, which is the value stored in a property attached to an object.

3.0 KNOWLEDGE BASE DESIGN

As previously stated, the critical task of this project is to design the knowledge base so that the expert system will have the ability to identify all impurities of interest that can be present in a silicon sample. These impurities are aluminum, boron, galium, indium, carbon, and oxygen. The presence of these impurities can be confirmed by identifying the variations they cause in a silicon absorption spectrum. This confirmation process involved three critical steps: (1) defining the wavenumber range where the peaks occur for each impurity, (2) identifying the patterns that confirm their existence, (3) and classifying all additional lines associated with that impurity which are not contained within the identifying pattern.

For the first step, the wavenumber ranges of the various impurities must be identified. From previous experimental knowledge, it is known that each possible impurity causes peaks to occur in a well defined region. For example, if aluminum is present in a silicon sample, peaks will appear in the spectrum in a range from 337 to 912 wavenumbers. As many as 20 different peaks can show up in this range. The problem lies in determining which peaks present in this range are actually due to the presence of aluminum and which ones are caused by extraneous noise.

This leads to the second step, identifying the patterns of each impurity to confirm its presence. Each of the 20 aluminum peaks that may appear can be distinctly defined by a particular wavenumber, which we will call *aluminum_line_n*. If aluminum is present, it is known from previous experimental knowledge that the pattern of aluminum_lines 1, 2, and 8 must appear in the spectrum. If strong identifiable peaks appear around 442 wavenumbers (aluminum_line_1), 471 wavenumbers (aluminum_line_2), and 543 wavenumbers (aluminum_line_8), then you can positively conclude that aluminum is present.

After this pattern matching has accomplished the identification of the impurity, the next step is to identify all additional peaks that are associated with that impurity. For aluminum, there may be as many as 17 peaks, since only 3 out of a possible 20 were needed to identify the pattern. These peaks must be identified by their wavenumbers and classified as aluminum_line_*n*. The important point to remember is that there may be peaks that show up at the aluminum_line_*n* wavenumbers, that are not due to the presence of aluminum. If the pattern of aluminum_lines 1, 2, and 8 has not been previously confirmed, the presence of these peaks must be caused by some extraneous noise. Any peaks that show up at the aluminum_line_*n* wavenumbers in the absence of the confirmation of the aluminum pattern would be considered unidentified.

In summary, the goal of this effort is to accomplish the impurity analysis for the six possible impurities of interest that can occur in silicon. These are aluminum, boron, gallium, indium, carbon and oxygen. The identification process is the same for each impurity. First, the wavenumber range where they will cause peaks to occur must be defined. Next, the patterns of peaks that confirm their existence must be identified. And finally, all additional peaks that are caused by the impurity which are not contained in the identifying pattern must be classified.

3.1 THE OBJECT NETWORK

As stated in the tutorial section, the object network represents the *representation dimension*. The representation is composed of classes, objects, and properties which can be thought of as symbolic representations of the data that the rules reason upon. The object network designed for this situation must therefore use classes, objects, and properties to represent the task of using spectral analysis to identify impurities in silicon.

The peaks that appear in a silicon spectrum that are due to the presence of impurities are classified as *lines*. Therefore a class will be used called LINES. It will have six subclasses for the six impurities that may be present, aluminum, boron, gallium, indium, carbon, and oxygen (Figure 4). These subclasses will be called AL_LINES, BO_LINES, GA_LINES, IN_LINES, CA_LINES, and OX_LINES. Classes are symbolically represented by circles.

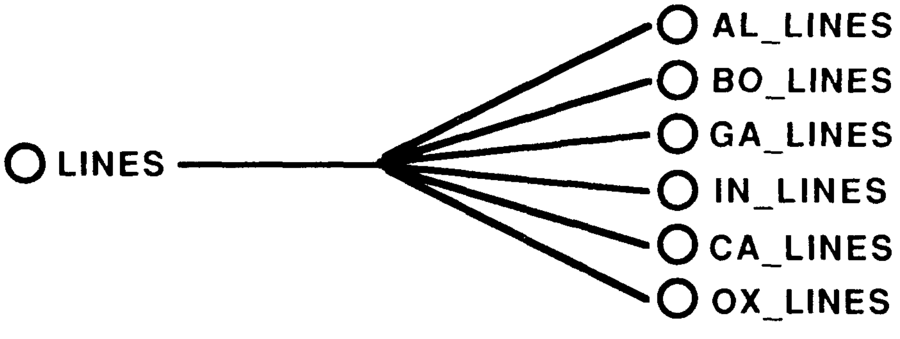


Figure 4. The class LINES

Each subclass will contain objects to represent all of the lines whose existence may be due to the presence of that impurity. For example, the subclass AL_LINES (Figure 5) will contain the objects AL_LINE_1, AL_LINE_2, etc. Each one of these objects has two properties, EXIST, and WN (wavenumber). EXIST will have a Boolean value and WN will have a numeric value. The slot AL_LINE_1.EXIST will be set to TRUE if that particular line exists. If the line exists, the slot AL_LINE_1.WN will be given the value of the wavenumber where the peak was found. Objects are symbolically represented by triangles and properties by squares.

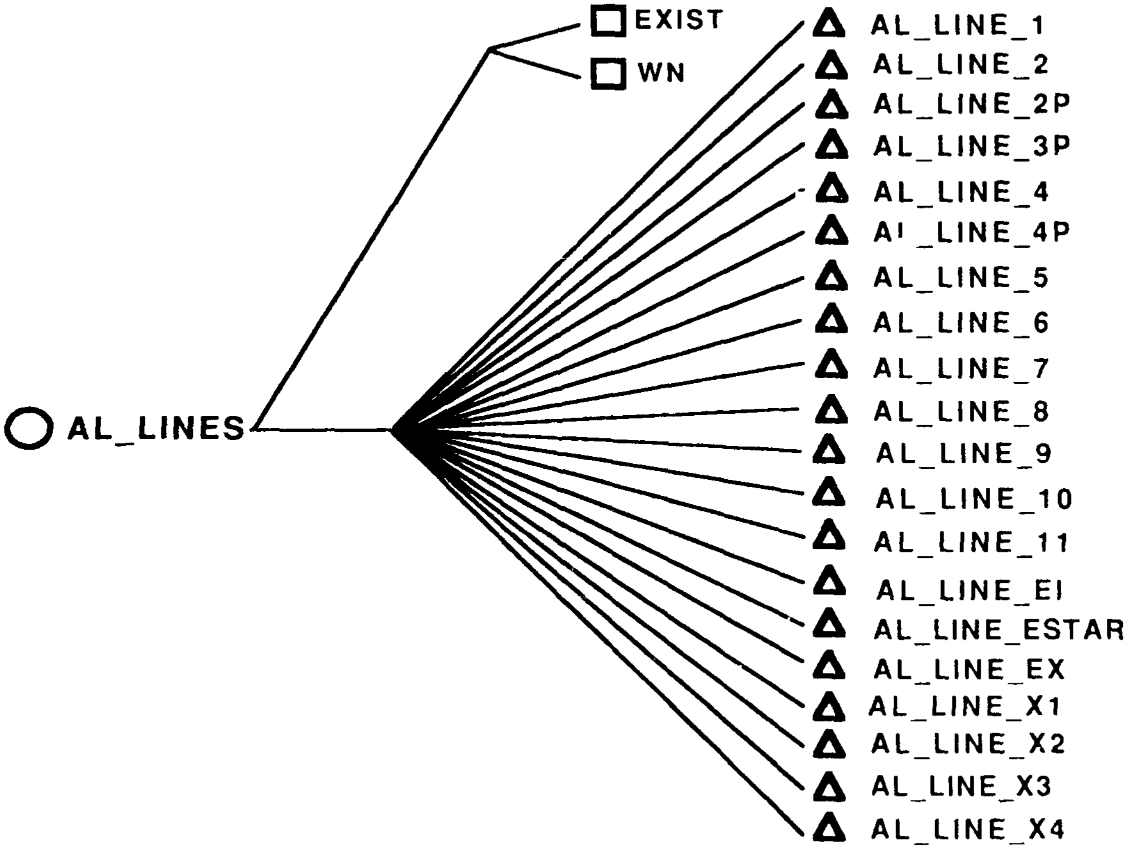


Figure 5. The class AL_LINES.

In the previous diagram, the properties EXIST and WN are shown attached to the class AL_LINES, rather than to the individual objects. This illustrates the principle of downward inheritance. If properties are initially attached to a class, then all the objects within that class will inherit these properties. This is very efficient for the situation where all of the objects in a class are similar items with the same characteristics. The properties EXIST and WN don't have to be individually attached to each of the 20 aluminum lines. As long as they are attached to the class, these properties will be inherited by all 20 objects which represent the aluminum lines.

Figure 6 shows the full extent of the class LINES. It is shown with all of its subclasses, and all of the objects that are contained within each subclass. As shown, downward inheritance also occurs from class to subclass. Initially, the properties EXIST and WN are attached to the class LINES. They are then inherited down by the subclasses AL_LINES, BO_LINES, etc., and from there they are inherited down to each individual object of all of the subclasses.

The objects illustrated represent all of the lines that the expert system will attempt to identify. Aluminum, boron, gallium, and indium are responsible for the majority of the impurity induced peaks that can occur in a silicon spectrum. Carbon and oxygen are responsible for causing a peak to occur at only one wavenumber each if they are present.

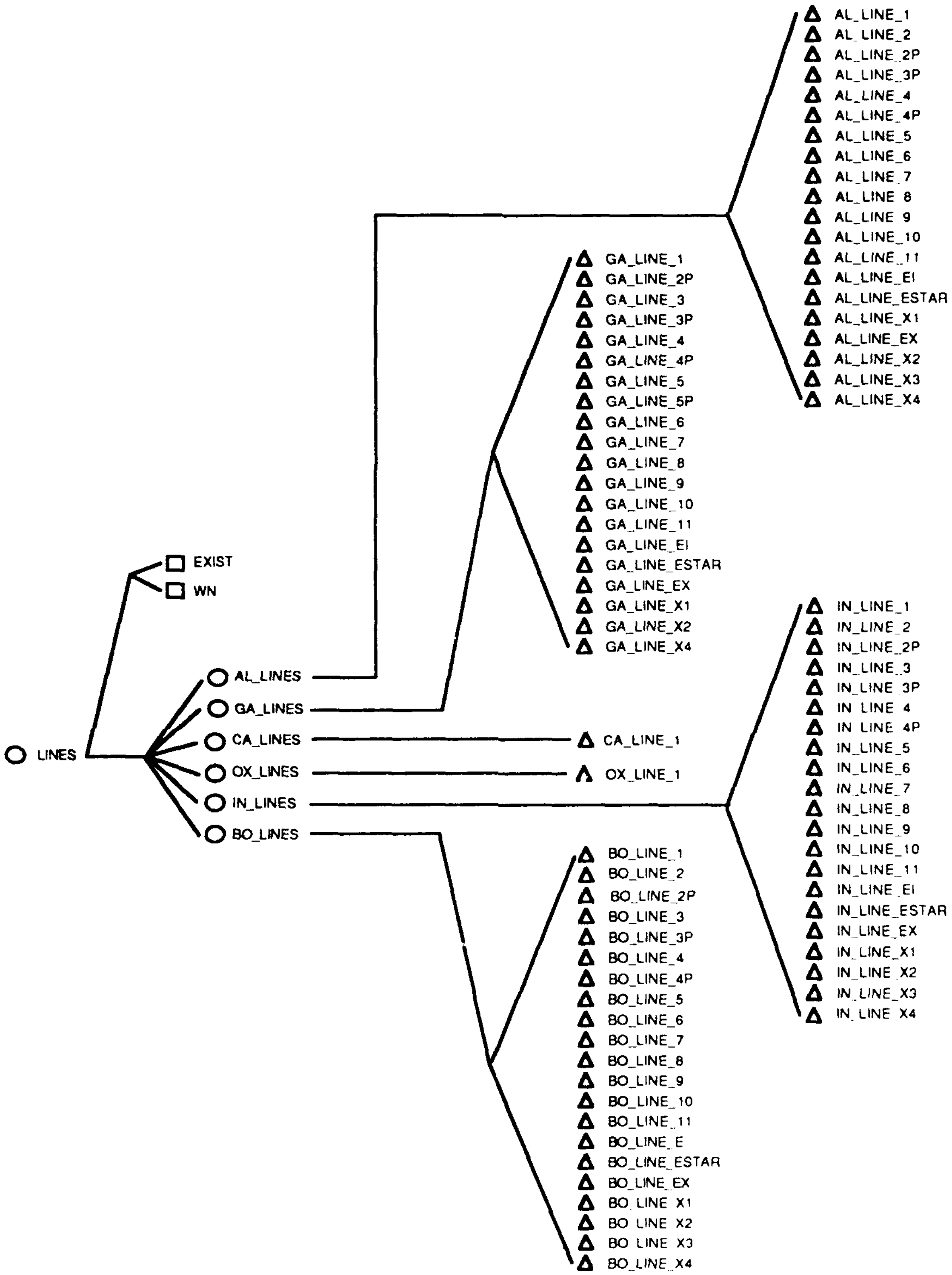


Figure 6. The class LINES, its properties, subclasses, and objects.

The presence of impurities in a sample of silicon may cause as many as 84 different peaks to appear in its spectrum. The class LINES contains six subclasses which contain all of the lines that are potentially caused by these impurities.

3.2 THE RULE NETWORK

The rule network represents the reasoning dimension. The reasoning dimension is composed of the rules that constitute the logic the expert system uses to perform reasoning. Rules were designed to categorize the peaks according to the wavenumber ranges of the impurities of interest, and also to identify their other lines if their patterns were recognized.

First, the expert system must obtain a list of the peaks that appeared on the silicon spectrum. After the spectrum has been generated on the FTIR spectrometer, a program is run which locates the peaks in the spectrum, and lists them in a table by wavenumber and amplitude. This peak table has the following format:

|name|	|wavenumber|	|amplitude|
Peak_1	432.56	0.654
Peak_2	458.78	0.421
Peak_3	543.89	0.784

If *n* peaks were found in the spectrum, *n* peaks will be listed in this peak table in ascending order by wavenumber. Peak_1 would have the lowest wavenumber and Peak_*n* the highest.

This peak table is read into the knowledge base. A class is created called POSSIBLE_PEAKS, and each peak is an object of this class (Figure 7). The name of the objects will be PEAK_1 through PEAK_*n*, and they will have the properties WAVENUMBER and AMPLITUDE. Using the previous table as an example, the slot PEAK_1.WAVENUMBER would contain the value 432.56, and the slot PEAK_1.AMPLITUDE would contain the value 0.654. With this symbolic representation of the initial peak data, the rules can now begin the reasoning process.

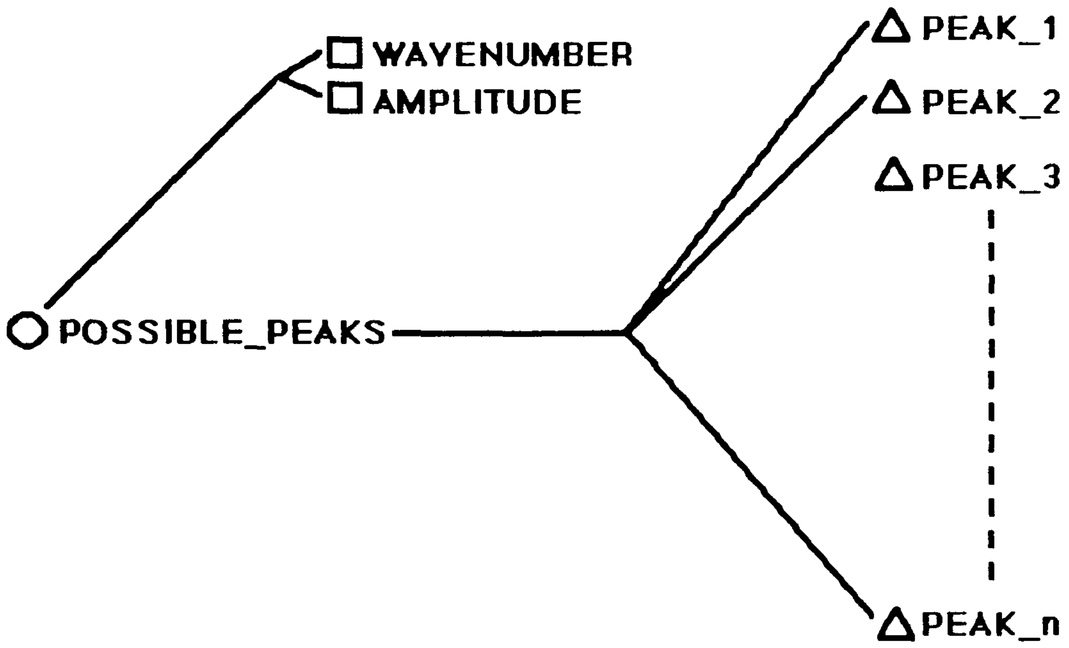


Figure 7. The class POSSIBLE_PEAKS.

The first task of the rule network is to categorize the peaks in terms of what impurity they might possibly represent. Since the wavelength range for aluminum peaks is 337 to 912 wavenumbers, a rule must be written that finds all of the peaks in the class

POSSIBLE_PEAKS that fall within this wavelength range. The name of this is PICK_POSSIBLE_ALUMINUM _PEAKS.

This rule will utilize a technique called pattern matching, which permits the creation of a list of objects that satisfy a certain set of conditions. First, the rule will sort through all the peaks in the class POSSIBLE_PEAKS and keep the ones whose wavenumber is greater than 337. Then it will sort through that list again and keep all of the peaks whose wavenumber is less than 912. Then it will take this final list and use it to create a new class called POSSIBLE _AL_PEAKS. This new class will contain all of the peaks whose peak location falls within the range of 337 to 912 wavenumbers. The rule looks as follows:

Name: PICK_POSSIBLE_ALUMINUM_PEAKS

Conditions:

```
    >        <|POSSIBLE_PEAKS|>.WAVENUMBER    337
    <        <|POSSIBLE_PEAKS|>.WAVENUMBER    912
```

Hypothesis: POSSIBLE_ALUMINUM

Actions:

```
CreateObject    <|POSSIBLE_PEAKS|>    |POSSIBLE_AL_PEAKS|
```

As seen above, rules are composed of three parts: *conditions*, a *hypothesis*, and *actions*. Conditions are conditional statements which must be evaluated to a Boolean true or false value. If all of the conditions evaluate to true, the hypothesis, which is also a Boolean value, will evaluate to true, and the actions will be performed. If even one of the conditions evaluates to false, the hypothesis will evaluate to false, and the actions will not be performed. In the preceding rule, there are two conditions and one action. If both of the conditions evaluate to true, the hypothesis POSSIBLE_ALUMINUM will evaluate to true, and the action will be performed. If the first condition evaluates to false, the hypothesis will automatically evaluate to false and the second will not even be evaluated.

The vertical bars enclosing POSSIBLE_PEAKS indicate that it is a class. The angle brackets outside the vertical bars indicate that pattern matching will occur. The attached property, WAVENUMBER, indicates that the WAVENUMBER slot of each object in the class POSSIBLE_PEAKS will be used for the comparison that is determined by the conditions.

The first condition tells the expert system to *pattern match*, which means to look at each object (PEAK_1 through PEAK_40) in the class POSSIBLE_PEAKS, and compare the value of the slot PEAK_*n*.WAVENUMBER to 337. If the value of this slot is greater than 337, the object PEAK_*n* is kept. If it finds at least one match, the condition will evaluate to true and a list will remain which contains all the objects whose slot PEAK_*n*.WAVENUMBER is greater than 337. If it doesn't find a match, the condition will evaluate to false, the hypothesis POSSIBLE_ ALUMINUM will be set to false and no further conditions will be evaluated. This would indicate that none of the peaks in the class POSSIBLE_PEAKS has a peak location wavenumber that is greater than 337, so they can't possibly be aluminum.

The second condition only gets evaluated if the first condition actually creates a list of objects whose PEAK_*n*.WAVENUMBER slot is greater than 337. It says to analyze this list of objects and keep the ones whose PEAK_*n*.WAVENUMBER slot is less than 912. If it finds at least one match, the condition will evaluate to true and a list will remain which contains all the objects whose PEAK_*n*.WAVE- NUMBER slot is greater than 337 and less than 912. If it doesn't find a match, this condition will evaluate to false which causes the hypothesis POSSIBLE_ALUMINUM to be set to false. This indicates that no peaks were found that fall in the wavenumber range established by the presence of aluminum.

If both conditions evaluate to true, the actions will be performed. A list of peaks will remain which might possibly be aluminum since they fall within its wavenumber range. Therefore, a new class will be created called POSSIBLE_AL_PEAKS. As seen in the example rule, this class will be created with the *CreateObject* command. The second term after the CreateObject command, |POSSIBLE_AL_PEAKS|, tells the expert system to create the new class. The first term after the CreateObject command, <|POSSIBLE_PEAKS|>, represents the list that was created by the first two conditions. So, the CreateObject command takes the list of objects that was compiled by virtue of the first two conditions and puts it into the new class POSSIBLE_AL_PEAKS.

For example, say the class POSSIBLE_PEAKS contains 100 peaks (PEAK_1 through PEAK_100), and the rule PICK_POSSIBLE_ALUMINUM_PEAKS found 15 peaks that fall within the aluminum wavenumber range. The class POSSIBLE_ AL_PEAKS would then contain 15 peaks (Figure 8).

The next step is to analyze the 15 peaks in the class POSSIBLE_AL_PEAKS and determine if the pattern exists which confirms the presence of aluminum. If three of these peaks have wavenumbers corresponding to aluminum lines 1, 2 and 8, then it can be concluded that aluminum is present. The wavenumbers for aluminum lines 1,2, and 8 are approximately 442.7, 471.9, and 443.9.

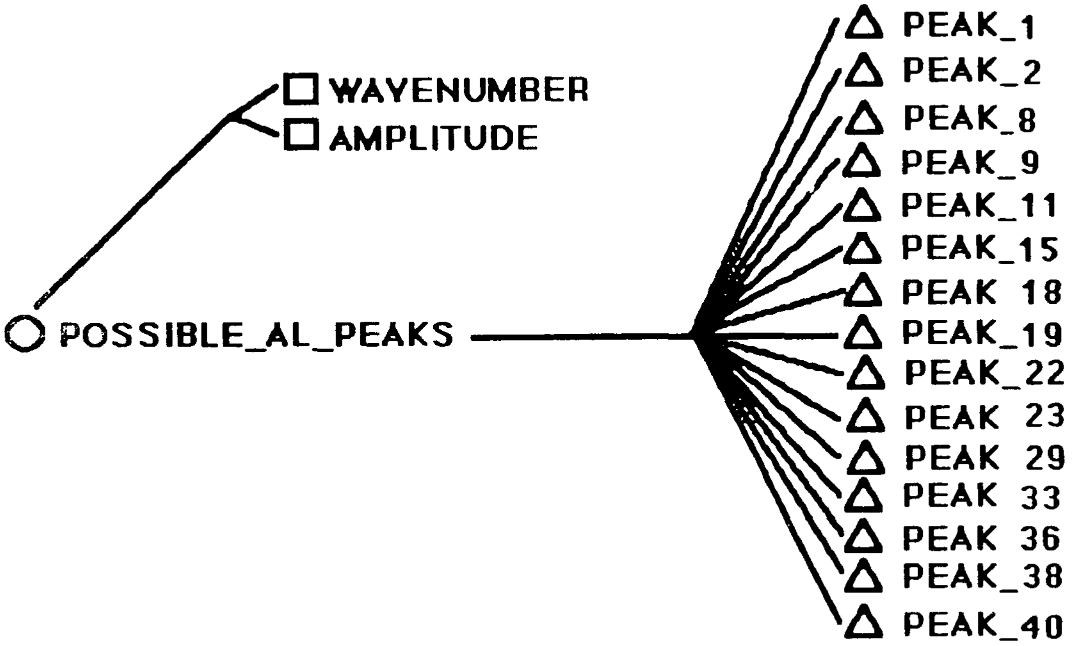


Figure 8. The class POSSIBLE_AL_PEAKS

A rule must be written that looks at the fifteen slots PEAK_n.WAVENUMBER to determine if there is a wavenumber match with aluminum lines 1, 2, and 8. The name of this rule will be INVESTIGATE_ALUMINUM. It appears on the following page.

The first condition evaluates the status of POSSIBLE_ALUMINUM, which is the hypothesis of the rule 'PICK_POSSIBLE_ALUMINUM_PEAKS.' The operator YES evaluates a Boolean value. This is equivalent to asking "Is the value of the hypothesis POSSIBLE_ALUMINUM true?" If POSSIBLE_ALUMINUM is true, this means that the previous rule found some peaks that might possibly be aluminum and stored them in the class POSSIBLE_AL_PEAKS. These peaks then need to be analyzed to see if the confirming pattern exists. This is accomplished with the rest of the conditions. If the value of the hypothesis POSSIBLE_ALUMINUM is false, no possible aluminum peaks were found, and the class POSSIBLE_AL_PEAKS was never

```
Name: INVESTIGATE_ALUMINUM
      Conditions:
         YES     POSSIBLE_ALUMINUM
          >     <|POSSIBLE_AL_PEAKS|>.WAVENUMBER           441.7
          <      <|POSSIBLE_AL_PEAKS|>.WAVENUMBER          443.7
          >     <<|POSSIBLE_AL_PEAKS|>>.WAVENUMBER         470.9
          <     <<|POSSIBLE_AL_PEAKS|>>.WAVENUMBER         472.9
         >     <<<|POSSIBLE_AL_PEAKS|>>>.WAVENUMBER        542.9
         <     <<<|POSSIBLE_AL_PEAKS|>>>.WAVENUMBER        544.9
      Hypothesis: ALUMINUM_PRESENT
      Actions:
         DO     <<|POSSIBLE_AL_PEAKS|>>.WAVENUMBER
                              AL_CALIBRATION_PEAK.WAVENUMBER
```

created. In this case, the hypothesis ALUMINUM_PRESENT will be automatically set to false, and the rest of the conditions will not be evaluated.

The next six conditions use pattern matching to determine if the aluminum confirmation pattern is present. The first two conditions sort through the list of objects in the class POSSIBLE_AL_PEAKS to see if a peak exists whose PEAK_*n*.WAVENUMBER slot is greater than 441.7 and less than 443.7. Since aluminum line 1 has a peak location of approximately 442.7 wavenumbers, this narrow interval will be sufficient to locate it if present. The next two conditions look for aluminum line 2, whose wavenumber is approximately 471.9. The double set of angle brackets tells the system to pattern match on the original list of peaks in the class POSSIBLE_AL_PEAKS. If only one set of angle brackets was used, this would mean "pattern match on the list created by the first two conditions," which contains only one peak. Since the second two conditions look for a completely different peak, they need to start over again with the original list of objects. The last two conditions employ the same methods to look for aluminum line 3, which has a wavenumber location of 543.9 wavenumbers. Once again, the triple set of angle brackets means to pattern match on the original list, not the list created by the second set of conditions with the double set of angle brackets.

If all the conditions evaluate to true, then the presence of aluminum is confirmed and the hypothesis ALUMINUM_PRESENT is set to true. If the hypothesis is true, the action will be performed. In this rule, the purpose of the action is to assign the value of the slot

AL_CALIBRATION_PEAK.WAVENUMBER the value contained in the PEAK_*n*.WAVENUMBER slot that was found with the pattern match that occurred in third and fourth conditions in the rule, which will be approximately 471.9 wavenumbers. This is indicated by the double angle brackets around the class |POSSIBLE_AL_PEAKS|. The 'DO' operator assigns the value of the second argument to the value of the first argument. So, after this action is completed, the slot AL_CALIBRATION_PEAK.WAVENUMBER will contain a value of approximately 471.9 wavenumbers.

This calibration peak is used to calculate the concentration of the impurity. Once the area of the calibration peak is known, it can be multiplied by a calibration factor to determine the concentration. Each impurity has its own calibration peak, and aluminum's happens to be the peak that occurs at approximately 471.9 wavenumbers. So, after the exact value of the AL_CALIBRATION_PEAK.WAVENUMBER slot is obtained, it is passed to the FORTRAN program 'FINDAREA' (see Appendix). This program uses this wavenumber to locate the peak in the raw spectral data that is stored in a common block. Once it locates the peak, it uses the trapezoidal integration method to determine its area. The area is then multiplied by the aluminum calibration factor to determine the aluminum concentration in the silicon sample.

If the aluminum confirmation pattern has been identified, the next step is to identify all other lines in the spectrum that are caused by the presence of aluminum. Aluminum may cause as many as 20 different peaks to appear, so 20 different rules are needed for their confirmation. Since the wavenumbers where these peaks occur is already known, pattern matching will be used on the class |POSSIBLE_ALUMINUM_PEAKS| to see if they exist. This same technique that was used to identify the aluminum confirmation pattern will now be used to identify the individual lines.

The first rule will be called ALUMINUM_LINE_1, and it will attempt to confirm the presence of the first aluminum line, which has a peak location of approximately 442.7 wavenumbers. The first thing this rule must do is check to see if the presence of aluminum has been confirmed. This is done by examining the value of the hypothesis ALUMINUM_PRESENT, which the previous rule set to true if the aluminum confirmation pattern was found. If ALUMINUM_PRESENT is true, then the rule must examine all peaks in the class |POSSIBLE_AL_PEAKS| to see if one has a peak location of 442.7 wavenumbers. If so, the slot AL_LINE_1.EXIST will be set to true and the slot AL_LINE_1.WN will be set to 442.7.

```
Name:  Aluminum_Line_1
Conditions:
Yes      ALUMINUM_PRESENT
 >      <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER   442
 <      <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER   443.5
Hypothesis:   ALUMINUM_LINE
Actions:
   Let      AL_LINE_1.EXIST       TRUE
  Do      <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER
                                                AL_LINE_1.WN
```

Twenty rules like this will exist for each possible aluminum line that may be present. For example, aluminum_line_X3 has a peak location of approximately 404 wavenumber. The rule to find this peak is:

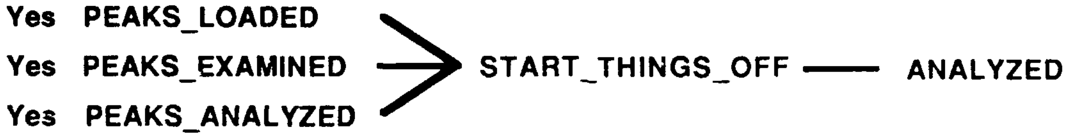
```
Name:  Aluminum_Line_X3
Conditions:
Yes      ALUMINUM_PRESENT
 >      <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER   403.5
 <      <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER   404.5
Hypothesis:   ALUMINUM_LINE
Actions:
   Let      AL_LINE_X3.EXIST      TRUE
  Do      <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER
                                            AL_LINE_X3.WN
```

3.3 PROGRAM FLOW

The previous chapters demonstrated how rules are structured and how they interact. Now it can be shown how these rules fit into the flow of the total program.

From now on, the rules will be shown in the way they appear in the rule network, which is a graphic display of the interrelation of all the rules. Looking at this 'rule tree' allows the developer to analyze the flow of the program. The previous rules were shown in the context of the rule editor, which is where they are created.

For example, the first rule the expert system evaluates is named START_THINGS_OFF:



This rule has three conditions on the left-hand side that must evaluate to TRUE before the hypothesis ANALYZED can be set to TRUE. These three conditions represent three separate branches of the rule network. The purpose of this rule is to force the evaluation of these three different branches.

The first branch is represented by the hypothesis PEAKS_LOADED (Figure 9). This branch of the network loads the spectrum and the peak table. After the spectrum and peak table have been loaded, the second branch of the network, represented by the hypothesis PEAKS_EXAMINED, is put into action (Figure 10). This branch of the network looks at the peak table and classifies the peaks into what they might possibly be, i.e., aluminum, boron, gallium, indium, carbon, oxygen, or unknown. The next branch of the network is represented by the hypothesis PEAKS_ANALYZED (Figure 11). This branch takes the categories of possible peaks that were just created and looks for the confirming patterns that tell which impurities are present. The following pages illustrate how these three branches are connected to the rule START_THINGS_OFF.

The three branches of the rule network interact in the following manner. The three conditions on the left hand side point to the rule name, START_THINGS_OFF, and the rule name points to the hypothesis ANALYZED. To get the process going, the hypothesis ANALYZED must be *suggested*. *Suggesting the hypothesis forces the* expert system to determine if its Boolean value is TRUE or FALSE. To do this, it must evaluate the conditions on the left-hand side. ANALYZED will be set to true if all three conditions evaluate to true. If any of the conditions evaluate to false, ANALYZED will be set to false.

In this rule, the left-hand side conditions involve three hypotheses to other rules. PEAKS_LOADED, PEAKS_EXAMINED, and PEAKS_ANALYZED are all hypotheses, and the *Yes* operator forces the evaluation of their Boolean value. When the expert system attempts to evaluate a hypothesis that appears in a condition, an event called *backward chaining* takes place which causes other rules in that branch of t he rule network to be evaluated. If the value of

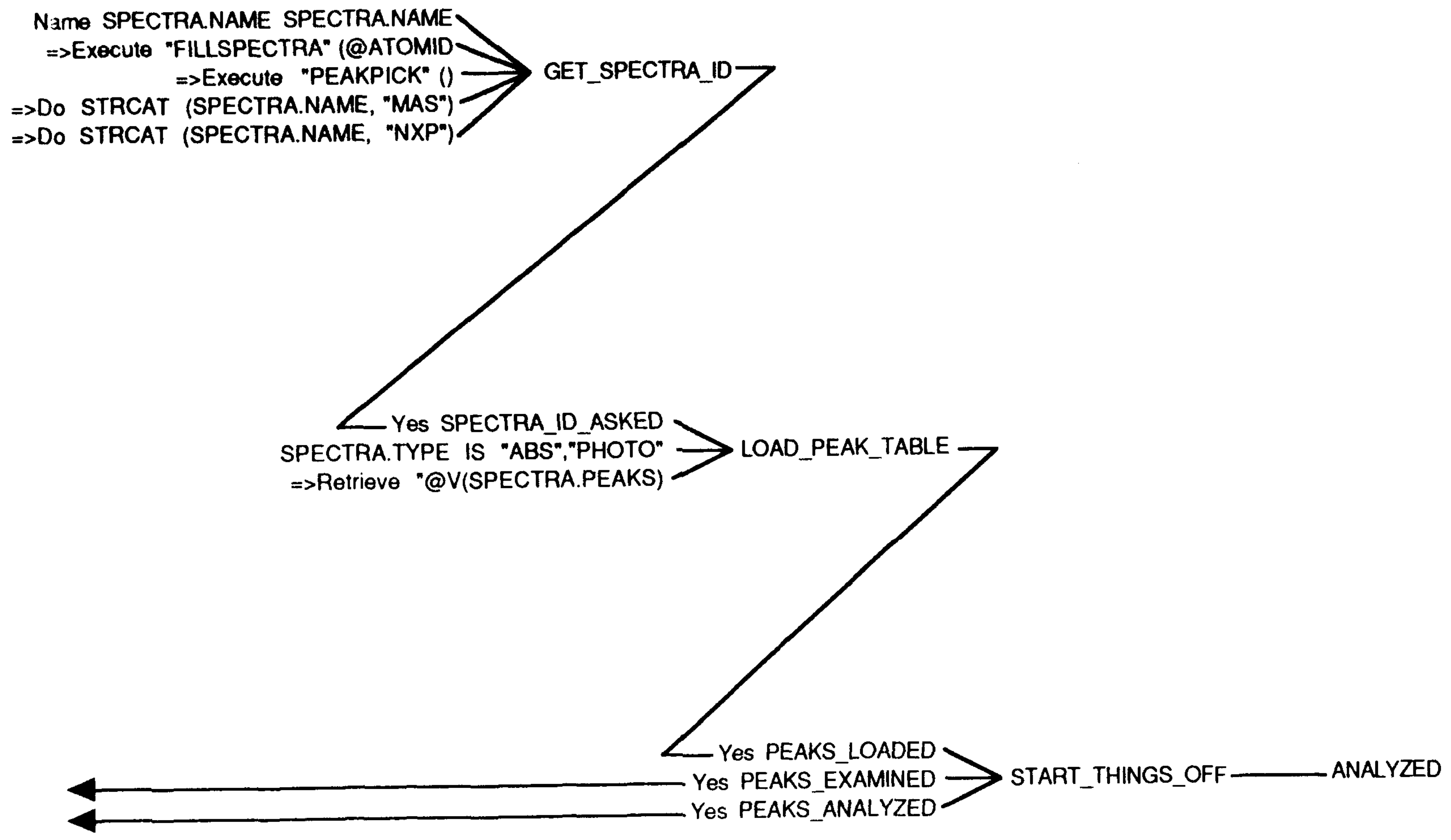

Name SPECTRA.NAME SPECTRA.NAME
=>Execute "FILLSPECTRA" (@ATOMID
=>Execute "PEAKPICK" ()
=>Do STRCAT (SPECTRA.NAME, "MAS")
=>Do STRCAT (SPECTRA.NAME, "NXP")
GET_SPECTRA_ID
Yes SPECTRA_ID_ASKED
SPECTRA.TYPE IS "ABS","PHOTO"
=>Retrieve "@V(SPECTRA.PEAKS)
LOAD_PEAK_TABLE
Yes PEAKS_LOADED
Yes PEAKS_EXAMINED
Yes PEAKS_ANALYZED
START_THINGS_OFF
ANALYZED

Figure 9. The rule START_THINGS_OFF and the branch of the rule network triggered by the hypothesis PEAKS_LOADED.

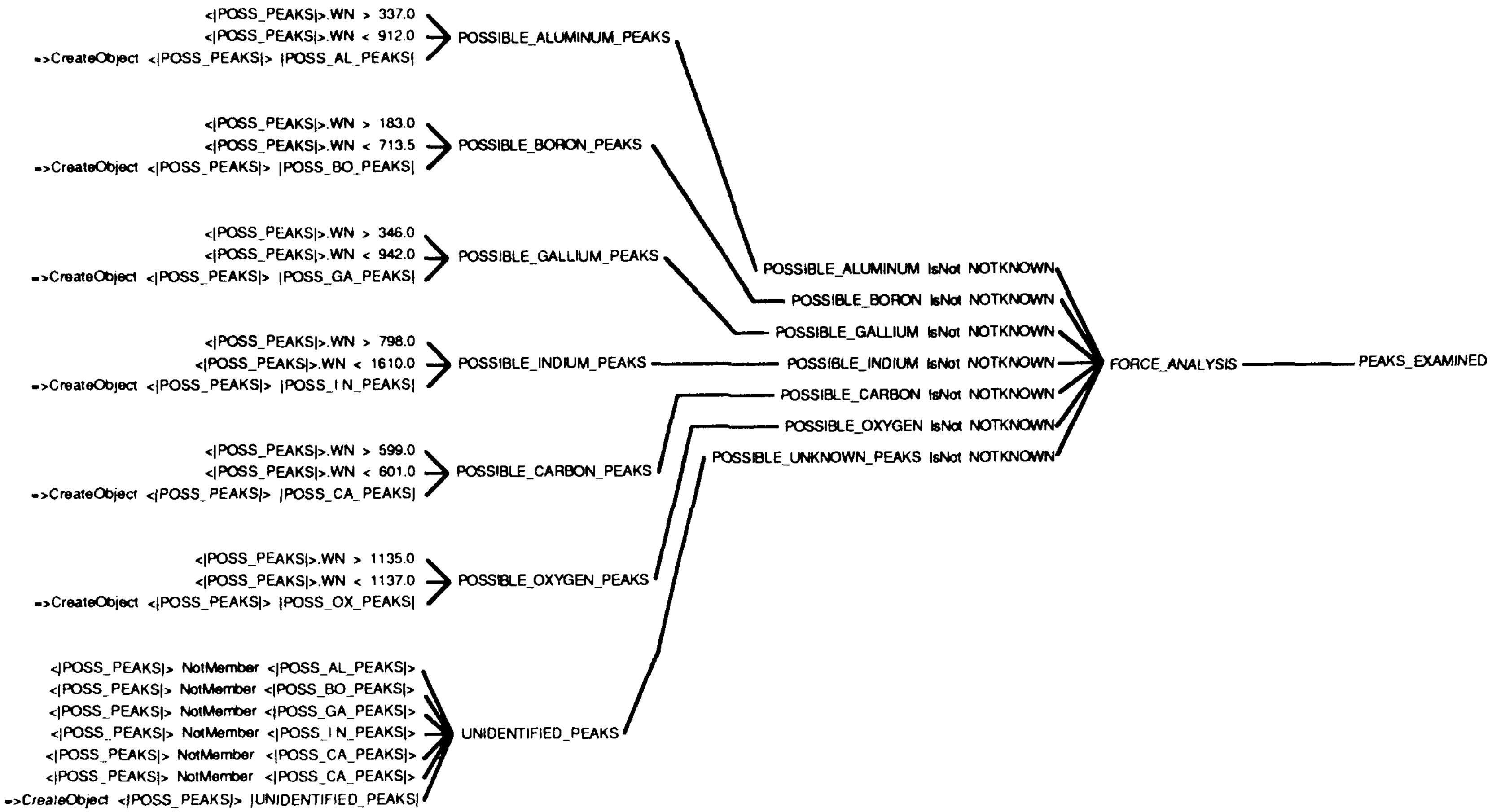

<|POSS_PEAKS|>.WN > 337.0
<|POSS_PEAKS|>.WN < 912.0
=>CreateObject <|POSS_PEAKS|> |POSS_AL_PEAKS|
POSSIBLE_ALUMINUM_PEAKS
<|POSS_PEAKS|>.WN > 183.0
<|POSS_PEAKS|>.WN < 713.5
=>CreateObject <|POSS_PEAKS|> |POSS_BO_PEAKS|
POSSIBLE_BORON_PEAKS
<|POSS_PEAKS|>.WN > 346.0
<|POSS_PEAKS|>.WN < 942.0
=>CreateObject <|POSS_PEAKS|> |POSS_GA_PEAKS|
POSSIBLE_GALLIUM_PEAKS
<|POSS_PEAKS|>.WN > 798.0
<|POSS_PEAKS|>.WN < 1610.0
=>CreateObject <|POSS_PEAKS|> |POSS_I N_PEAKS|
POSSIBLE_INDIUM_PEAKS
<|POSS_PEAKS|>.WN > 599.0
<|POSS_PEAKS|>.WN < 601.0
=>CreateObject <|POSS_PEAKS|> |POSS_CA_PEAKS|
POSSIBLE_CARBON_PEAKS
<|POSS_PEAKS|>.WN > 1135.0
<|POSS_PEAKS|>.WN < 1137.0
=>CreateObject <|POSS_PEAKS|> |POSS_OX_PEAKS|
POSSIBLE_OXYGEN_PEAKS
<|POSS_PEAKS|> NotMember <|POSS_AL_PEAKS|>
<|POSS_PEAKS|> NotMember <|POSS_BO_PEAKS|>
<|POSS_PEAKS|> NotMember <|POSS_GA_PEAKS|>
<|POSS_PEAKS|> NotMember <|POSS_I N_PEAKS|>
<|POSS_PEAKS|> NotMember <|POSS_CA_PEAKS|>
<|POSS_PEAKS|> NotMember <|POSS_CA_PEAKS|>
=>CreateObject <|POSS_PEAKS|> |UNIDENTIFIED_PEAKS|
UNIDENTIFIED_PEAKS
POSSIBLE_ALUMINUM IsNot NOTKNOWN
POSSIBLE_BORON IsNot NOTKNOWN
POSSIBLE_GALLIUM IsNot NOTKNOWN
POSSIBLE_INDIUM IsNot NOTKNOWN
POSSIBLE_CARBON IsNot NOTKNOWN
POSSIBLE_OXYGEN IsNot NOTKNOWN
POSSIBLE_UNKNOWN_PEAKS IsNot NOTKNOWN
FORCE_ANALYSIS
PEAKS_EXAMINED

Figure 10. The PEAKS_EXAMINED branch of the rule network.

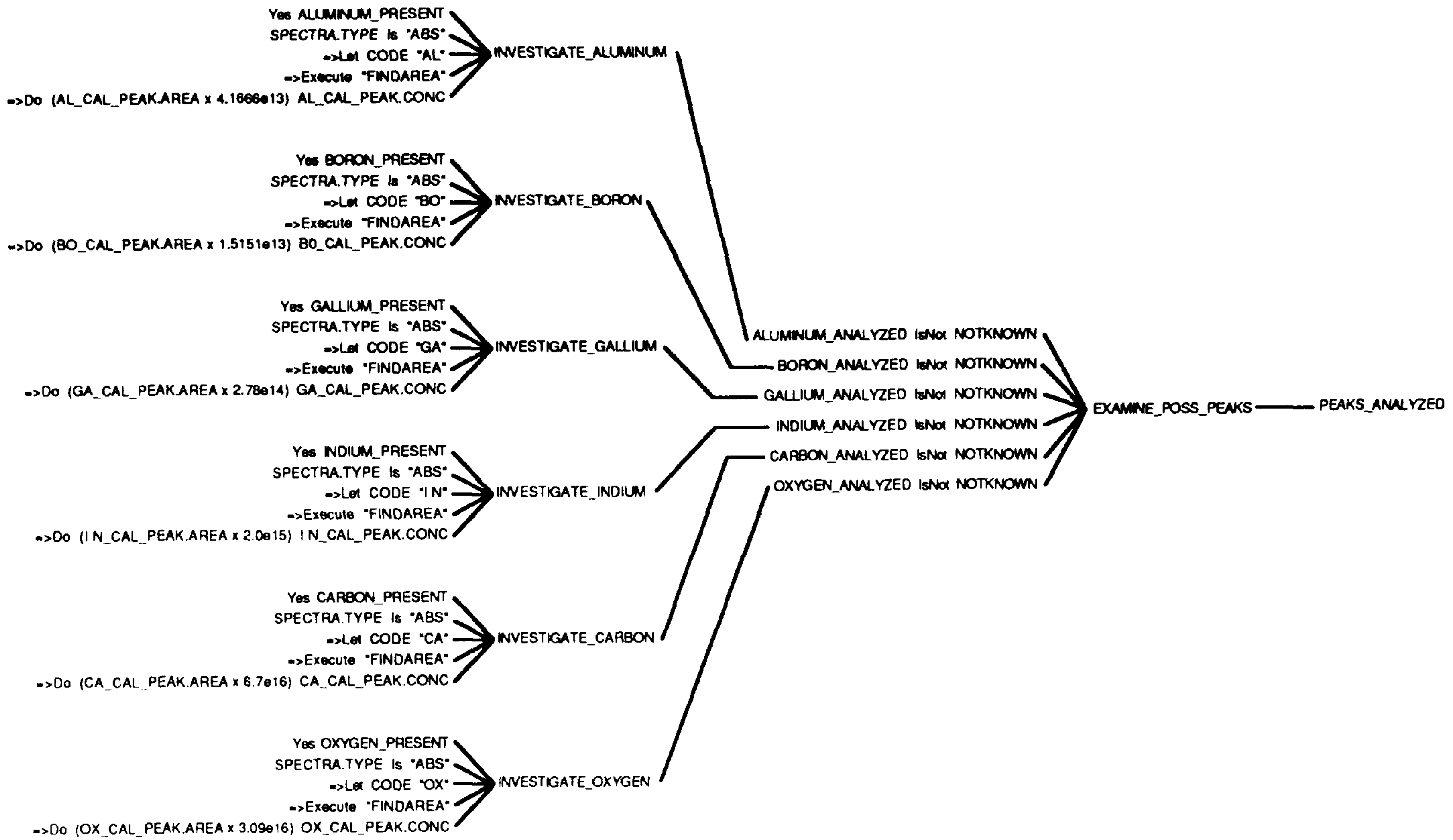


Figure 11. The PEAKS_ANALYZED branch of the rule network.

the hypothesis is unknown, then the rule leading to that hypothesis must be evaluated.

Since all the rules in a branch are connected and dependent on each other, trying to evaluate one rule will cause all rules leading up to it to also be evaluated. For example, the first condition in the rule START_THINGS_OFF attempts to determine the value of the hypothesis PEAKS_LOADED, which doesn't yet have a value. To determine its value, the expert system must evaluate the rule LOAD_PEAK_TABLE, since this rule has PEAKS_LOADED as its hypothesis. If all the conditions on the left-hand side of the rule LOAD_PEAK_TABLE evaluate to true, then the hypothesis PEAKS_LOADED will be set to true.

However, the first condition of the rule LOAD_PEAK_TABLE attempts to evaluate the value of SPECTRA_ID_ ASKED which is the hypothesis of the rule GET_SPECTRA_ID. So the rule GET_SPECTRA_ID must be evaluated before the hypothesis SPECTRA_ID_ASKED will have a value. Since none of the conditions in the rule GET_SPECTRA_ID attempt to evaluate the hypothesis of another rule, this is the end of the 'backward chain.'

To summarize, 'suggesting' the hypothesis ANALYZED forced the evaluation of the rule START_THINGS_OFF. The first condition in this rule, 'Yes PEAKS_LOADED', forced the evaluation of the rule LOAD_PEAK_TABLE. The first condition of this rule, 'Yes SPECTRA_ID_ASKED', forced the evaluation of the rule GET_SPECTRA_ID. Backward chaining forced the evaluation of all the rules in the PEAKS_LOADED branch of the rule network.

After backward chaining forces the evaluation of the three branches of the rule network, ANALYZED will then be set to true. At this point, the peak table has been loaded. It has been examined so that the peaks are classified as to what impurity they might possibly represent, and the groups of possible peaks have been analyzed to see if the confirmation pattern for the various impurities exist.

If it has been determined that the confirmation pattern exists for one or more impurities, the next step is to identify all the peaks that exist due to their presence. This is accomplished with six more rules, which are named AL_LINES, BO_LINES, GA_LINES, IN_LINES, CA_LINES, and OX_LINES. For example, if aluminum is present, then the rule AL_LINES will be evaluated to identify all of the aluminum peaks. This rule will not be evaluated if aluminum is not present. There are approximately 20 different peaks that can appear for each impurity, and these rules will identify any of these peaks that may be present.

These rules are evaluated with a process called *forward chaining*, which is the opposite of backward chaining. In backward chaining, when a hypothesis appears as data in a left-hand side condition of some initial rule, the expert system goes backward through the rule network to evaluate all rules that are needed to determine the value of this hypothesis. In forward chaining, once the value of a hypothesis or any other data value is determined, this value is volunteered forward to any other rules that uses it. Any time the expert system gives a rule a data value it needs to evaluate, the rule will automatically get evaluated without having to suggest its hypothesis.

Figure 12 illustrates the concept of forward chaining. The previous example showed how three different branches of the rule network branched backwards from the rule START_THINGS_OFF. This example shows how another section of the rule network branches forward from the same rule. The six rules that identify the peaks of the six impurities depend on the value of the hypothesis ANALYZED of the rule START_THINGS_ OFF. This illustration shows how these six rules branch forward from the hypothesis ANALYZED.

After the rule START_THINGS_OFF has been successfully evaluated, the value for ANALYZED will be TRUE. This value will then be volunteered to the six rules AL_LINES, BO_LINES, GA_LINES, IN_LINES, CA_LINES, and OX_LINES. If, for example, the presence of aluminun has been confirmed, the value of the hypothesis ALUMINUM_PRESENT will be TRUE. ALUMINUM_PRESENT is the hypothesis of the rule INVESTIGATE_ALUMINUM, which looked for presence of the aluminum confirmation pattern. So, looking at the rule AL_LINES, the first condition 'Yes ANALYZED' will be true, the second condition 'Yes ALUMINUM_PRESENT' will be true, therefore the hypothesis FIND_AL_LINES will evaluate to TRUE, so the action statement 'Do ALUMINUM_LINE ALUMINUM_LINE will be performed.

ALUMINUM_LINE is a hypothesis, so the action statement 'Do ALUMINUM_LINE ALUMINUM_LINE' is the same as 'suggesting' this hypothesis. This simply forces the evaluation of any rule that has ALUMINUM_LINE as its hypothesis. In Figure 13 there are twenty rules that have ALUMINUM_LINE as its hypothesis.

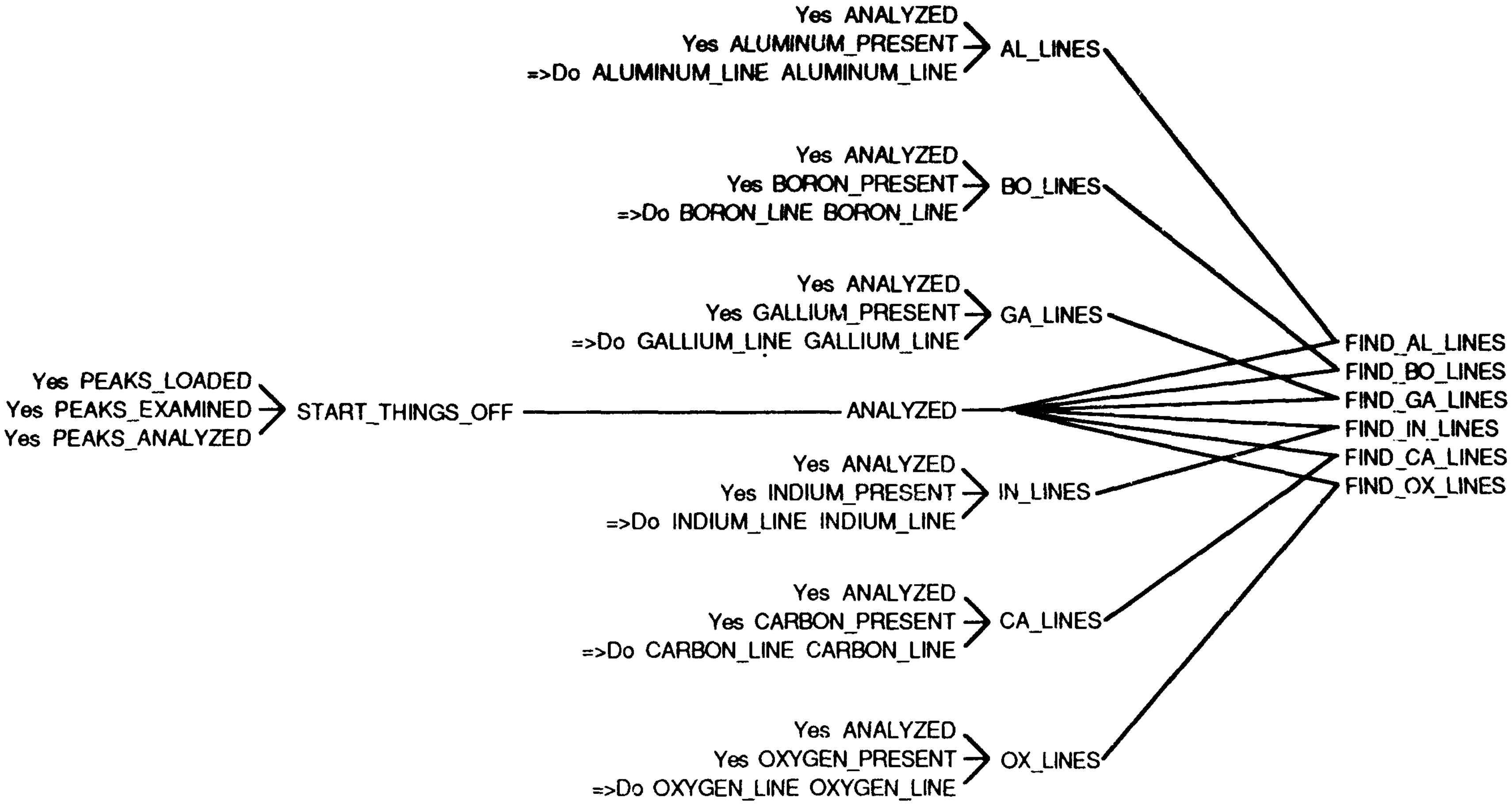


Figure 12. Forward chaining from the hypothesis ANALYZED.

Each one of these rules tries to identify a particular peak and sets the value of the slot AL_LINE_*n*.EXIST to TRUE if that peak is found. It also sets the value of AL_LINE_*n*.WN to the wavenumber of the peak location. So, the action statement 'Do ALUMINUM_LINE ALUMINUM_LINE' triggers the evaluation of 20 rules which will positively identify all of the peaks that are in the spectrum due to the presence of aluminum.

Following the rules that identify the aluminum lines are the rules that identify the lines for all the other impurities (Figures 14 through 17). It is not possible to show all of the rules for each of the impurities. For the rules that are not shown in their entirety, their peak locations can be found in Table 2.

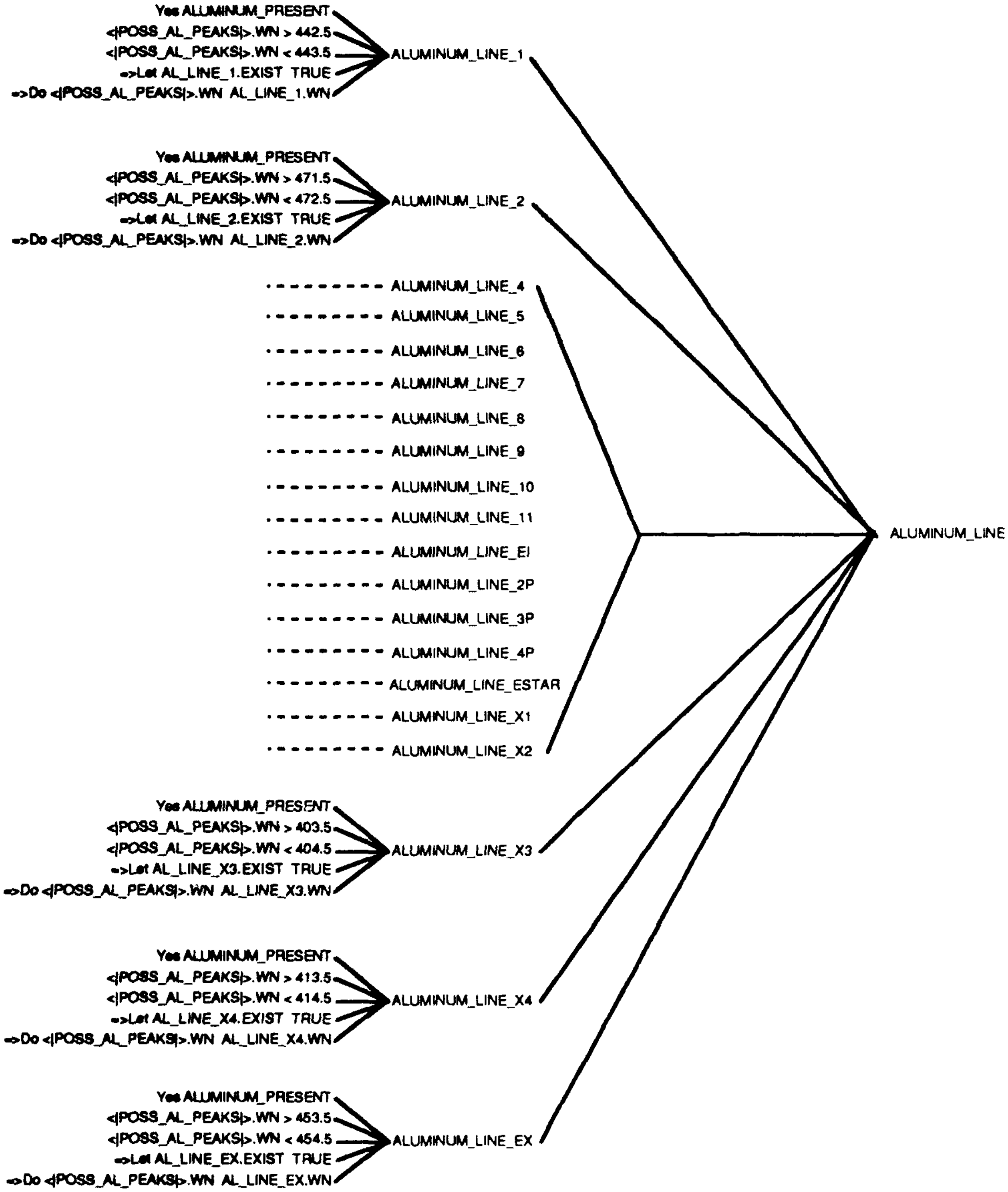

Yes ALUMINUM_PRESENT
<|POSS_AL_PEAKS|>.WN > 442.5
<|POSS_AL_PEAKS|>.WN < 443.5
=>Let AL_LINE_1.EXIST TRUE
=>Do <|POSS_AL_PEAKS|>.WN AL_LINE_1.WN
ALUMINUM_LINE_1
Yes ALUMINUM_PRESENT
<|POSS_AL_PEAKS|>.WN > 471.5
<|POSS_AL_PEAKS|>.WN < 472.5
=>Let AL_LINE_2.EXIST TRUE
=>Do <|POSS_AL_PEAKS|>.WN AL_LINE_2.WN
ALUMINUM_LINE_2
ALUMINUM_LINE_4
ALUMINUM_LINE_5
ALUMINUM_LINE_6
ALUMINUM_LINE_7
ALUMINUM_LINE_8
ALUMINUM_LINE_9
ALUMINUM_LINE_10
ALUMINUM_LINE_11
ALUMINUM_LINE_EI
ALUMINUM_LINE_2P
ALUMINUM_LINE_3P
ALUMINUM_LINE_4P
ALUMINUM_LINE_ESTAR
ALUMINUM_LINE_X1
ALUMINUM_LINE_X2
ALUMINUM_LINE
Yes ALUMINUM_PRESENT
<|POSS_AL_PEAKS|>.WN > 403.5
<|POSS_AL_PEAKS|>.WN < 404.5
=>Let AL_LINE_X3.EXIST TRUE
=>Do <|POSS_AL_PEAKS|>.WN AL_LINE_X3.WN
ALUMINUM_LINE_X3
Yes ALUMINUM_PRESENT
<|POSS_AL_PEAKS|>.WN > 413.5
<|POSS_AL_PEAKS|>.WN < 414.5
=>Let AL_LINE_X4.EXIST TRUE
=>Do <|POSS_AL_PEAKS|>.WN AL_LINE_X4.WN
ALUMINUM_LINE_X4
Yes ALUMINUM_PRESENT
<|POSS_AL_PEAKS|>.WN > 453.5
<|POSS_AL_PEAKS|>.WN < 454.5
=>Let AL_LINE_EX.EXIST TRUE
=>Do <|POSS_AL_PEAKS|>.WN AL_LINE_EX.WN
ALUMINUM_LINE_EX

Figure 13. The aluminum rules.

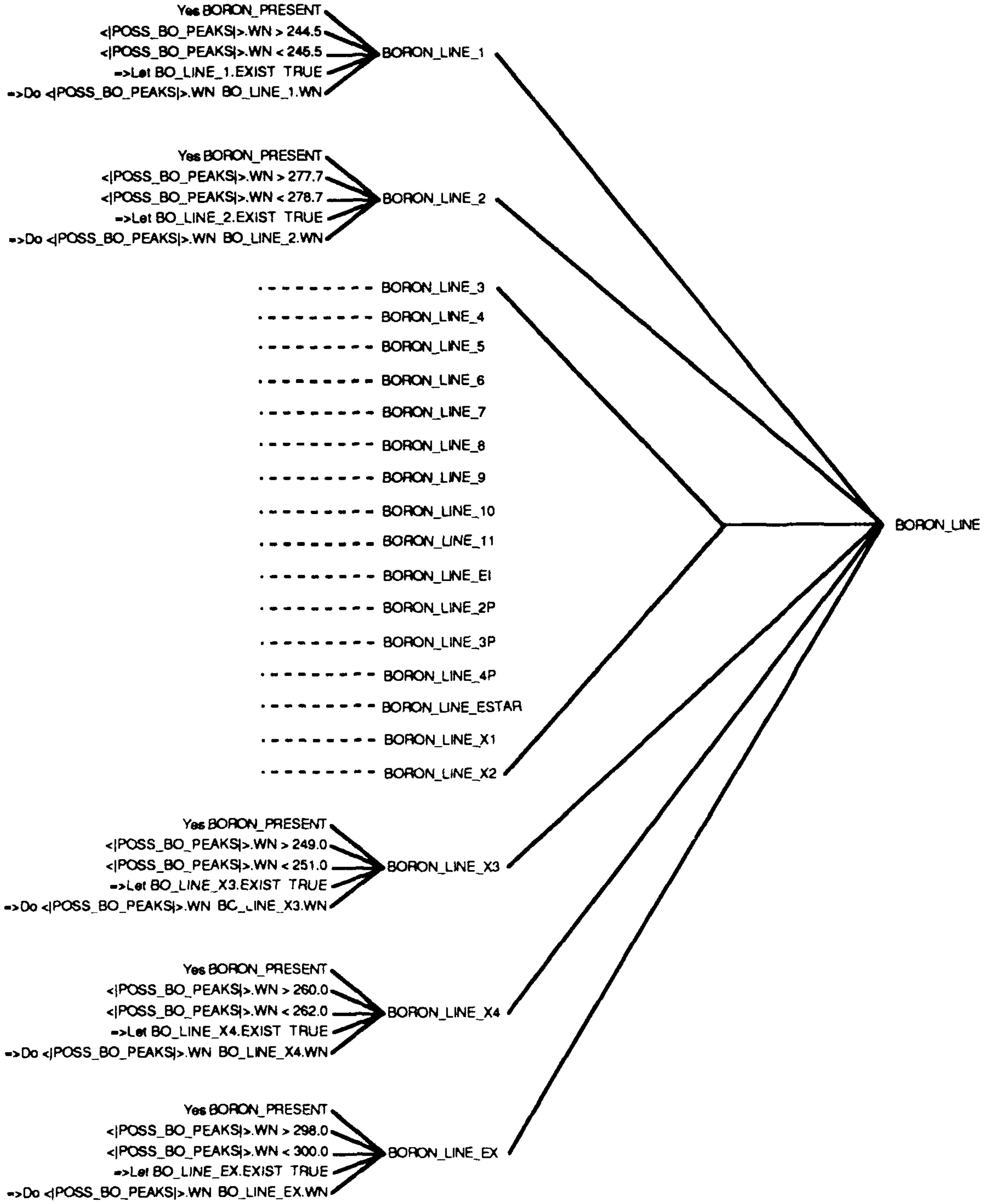

Yes BORON_PRESENT
<|POSS_BO_PEAKS|>.WN > 244.5
<|POSS_BO_PEAKS|>.WN < 245.5
=>Let BO_LINE_1.EXIST TRUE
=>Do <|POSS_BO_PEAKS|>.WN BO_LINE_1.WN
BORON_LINE_1
Yes BORON_PRESENT
<|POSS_BO_PEAKS|>.WN > 277.7
<|POSS_BO_PEAKS|>.WN < 278.7
=>Let BO_LINE_2.EXIST TRUE
=>Do <|POSS_BO_PEAKS|>.WN BO_LINE_2.WN
BORON_LINE_2
BORON_LINE_3
BORON_LINE_4
BORON_LINE_5
BORON_LINE_6
BORON_LINE_7
BORON_LINE_8
BORON_LINE_9
BORON_LINE_10
BORON_LINE_11
BORON_LINE_EI
BORON_LINE_2P
BORON_LINE_3P
BORON_LINE_4P
BORON_LINE_ESTAR
BORON_LINE_X1
BORON_LINE_X2
Yes BORON_PRESENT
<|POSS_BO_PEAKS|>.WN > 249.0
<|POSS_BO_PEAKS|>.WN < 251.0
=>Let BO_LINE_X3.EXIST TRUE
=>Do <|POSS_BO_PEAKS|>.WN BC_.INE_X3.WN
BORON_LINE_X3
Yes BORON_PRESENT
<|POSS_BO_PEAKS|>.WN > 260.0
<|POSS_BO_PEAKS|>.WN < 262.0
=>Let BO_LINE_X4.EXIST TRUE
=>Do <|POSS_BO_PEAKS|>.WN BO_LINE_X4.WN
BORON_LINE_X4
Yes BORON_PRESENT
<|POSS_BO_PEAKS|>.WN > 298.0
<|POSS_BO_PEAKS|>.WN < 300.0
=>Let BO_LINE_EX.EXIST TRUE
=>Do <|POSS_BO_PEAKS|>.WN BO_LINE_EX.WN
BORON_LINE_EX
BORON_LINE

Figure 14. The boron rules.

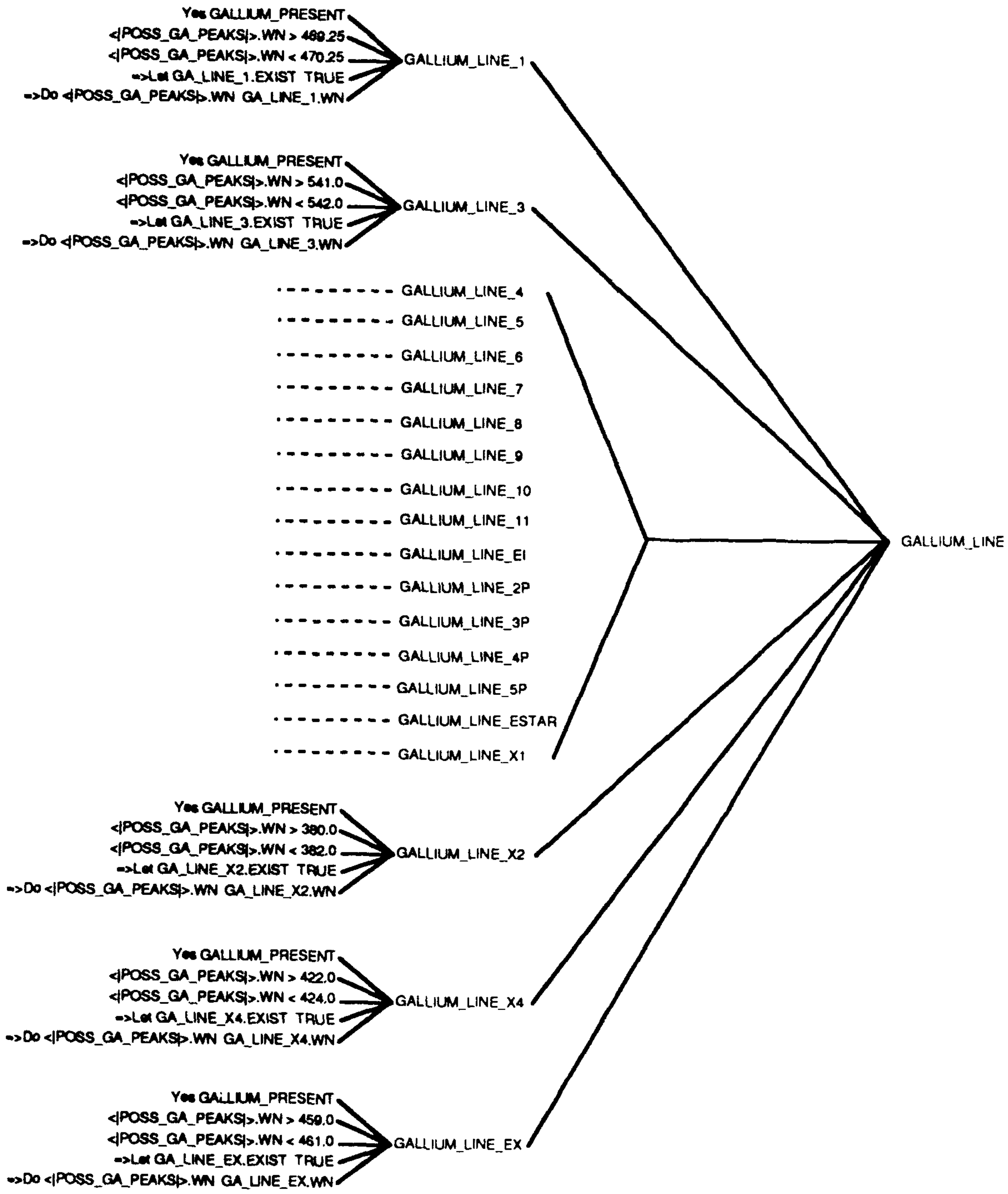

Yes GALLIUM_PRESENT
<|POSS_GA_PEAKS|>.WN > 469.25
<|POSS_GA_PEAKS|>.WN < 470.25
=>Let GA_LINE_1.EXIST TRUE
=>Do <|POSS_GA_PEAKS|>.WN GA_LINE_1.WN
GALLIUM_LINE_1
Yes GALLIUM_PRESENT
<|POSS_GA_PEAKS|>.WN > 541.0
<|POSS_GA_PEAKS|>.WN < 542.0
=>Let GA_LINE_3.EXIST TRUE
=>Do <|POSS_GA_PEAKS|>.WN GA_LINE_3.WN
GALLIUM_LINE_3
GALLIUM_LINE_4
GALLIUM_LINE_5
GALLIUM_LINE_6
GALLIUM_LINE_7
GALLIUM_LINE_8
GALLIUM_LINE_9
GALLIUM_LINE_10
GALLIUM_LINE_11
GALLIUM_LINE_EI
GALLIUM_LINE_2P
GALLIUM_LINE_3P
GALLIUM_LINE_4P
GALLIUM_LINE_5P
GALLIUM_LINE_ESTAR
GALLIUM_LINE_X1
GALLIUM_LINE
Yes GALLIUM_PRESENT
<|POSS_GA_PEAKS|>.WN > 380.0
<|POSS_GA_PEAKS|>.WN < 382.0
=>Let GA_LINE_X2.EXIST TRUE
=>Do <|POSS_GA_PEAKS|>.WN GA_LINE_X2.WN
GALLIUM_LINE_X2
Yes GALLIUM_PRESENT
<|POSS_GA_PEAKS|>.WN > 422.0
<|POSS_GA_PEAKS|>.WN < 424.0
=>Let GA_LINE_X4.EXIST TRUE
=>Do <|POSS_GA_PEAKS|>.WN GA_LINE_X4.WN
GALLIUM_LINE_X4
Yes GALLIUM_PRESENT
<|POSS_GA_PEAKS|>.WN > 459.0
<|POSS_GA_PEAKS|>.WN < 461.0
=>Let GA_LINE_EX.EXIST TRUE
=>Do <|POSS_GA_PEAKS|>.WN GA_LINE_EX.WN
GALLIUM_LINE_EX

Figure 15. The gallium rules.

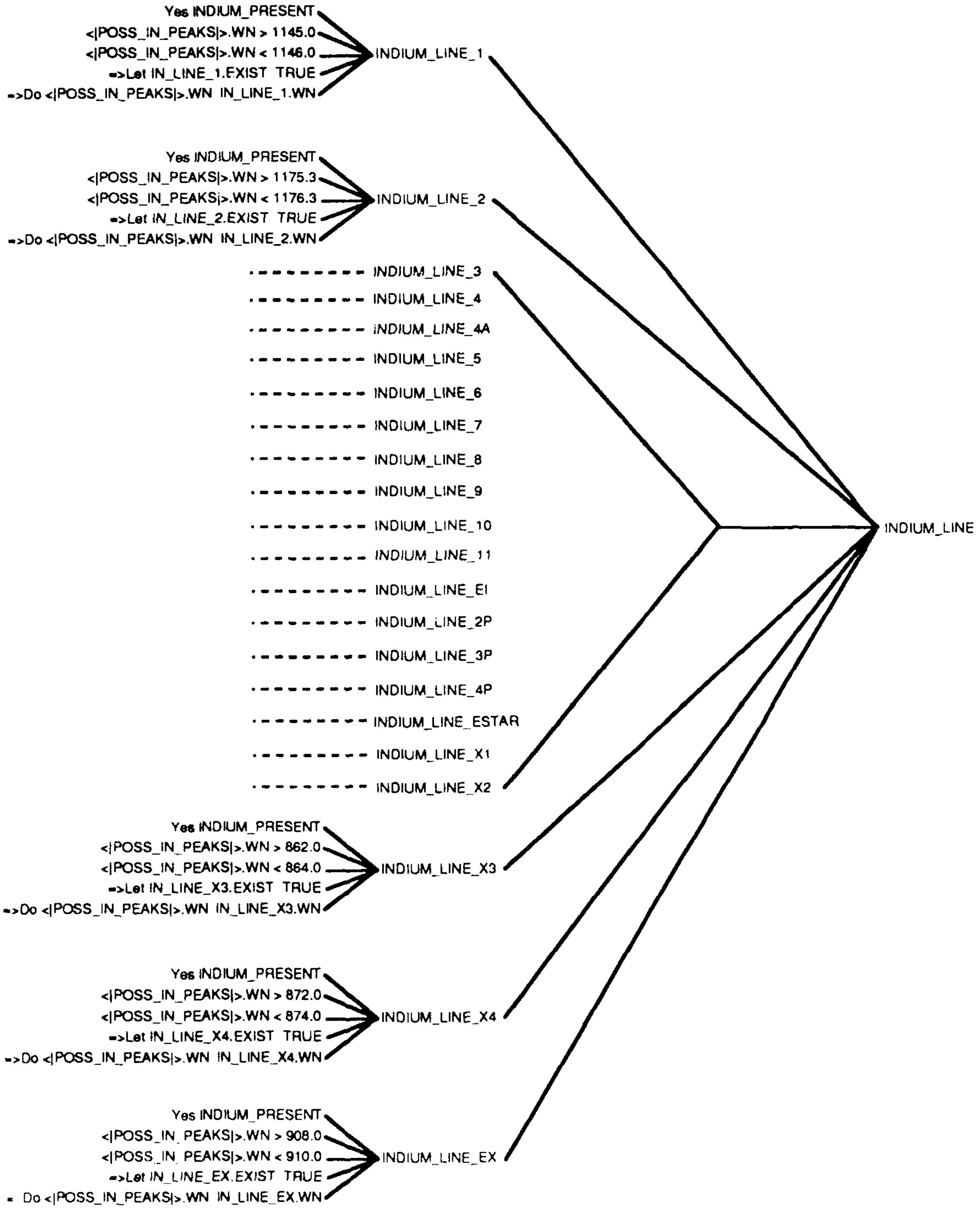

Yes INDIUM_PRESENT
<|POSS_IN_PEAKS|>.WN > 1145.0
<|POSS_IN_PEAKS|>.WN < 1146.0
=>Let IN_LINE_1.EXIST TRUE
=>Do <|POSS_IN_PEAKS|>.WN IN_LINE_1.WN
INDIUM_LINE_1
Yes INDIUM_PRESENT
<|POSS_IN_PEAKS|>.WN > 1175.3
<|POSS_IN_PEAKS|>.WN < 1176.3
=>Let IN_LINE_2.EXIST TRUE
=>Do <|POSS_IN_PEAKS|>.WN IN_LINE_2.WN
INDIUM_LINE_2
INDIUM_LINE_3
INDIUM_LINE_4
INDIUM_LINE_4A
INDIUM_LINE_5
INDIUM_LINE_6
INDIUM_LINE_7
INDIUM_LINE_8
INDIUM_LINE_9
INDIUM_LINE_10
INDIUM_LINE_11
INDIUM_LINE_EI
INDIUM_LINE_2P
INDIUM_LINE_3P
INDIUM_LINE_4P
INDIUM_LINE_ESTAR
INDIUM_LINE_X1
INDIUM_LINE_X2
INDIUM_LINE
Yes INDIUM_PRESENT
<|POSS_IN_PEAKS|>.WN > 862.0
<|POSS_IN_PEAKS|>.WN < 864.0
=>Let IN_LINE_X3.EXIST TRUE
=>Do <|POSS_IN_PEAKS|>.WN IN_LINE_X3.WN
INDIUM_LINE_X3
Yes INDIUM_PRESENT
<|POSS_IN_PEAKS|>.WN > 872.0
<|POSS_IN_PEAKS|>.WN < 874.0
=>Let IN_LINE_X4.EXIST TRUE
=>Do <|POSS_IN_PEAKS|>.WN IN_LINE_X4.WN
INDIUM_LINE_X4
Yes INDIUM_PRESENT
<|POSS_IN_PEAKS|>.WN > 908.0
<|POSS_IN_PEAKS|>.WN < 910.0
=>Let IN_LINE_EX.EXIST TRUE
= Do <|POSS_IN_PEAKS|>.WN IN_LINE_EX.WN
INDIUM_LINE_EX

Figure 16. The indium rules.

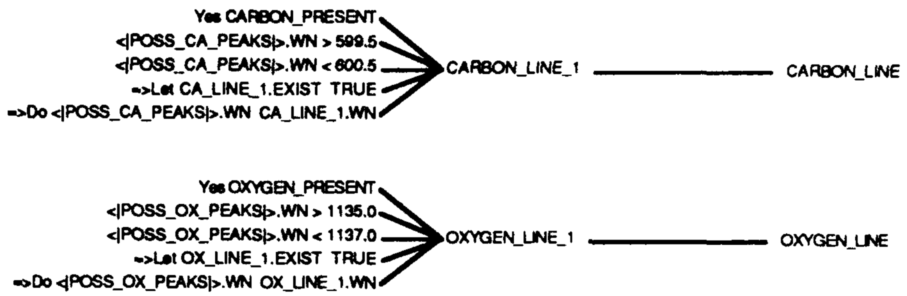

Yes CARBON_PRESENT
<|POSS_CA_PEAKS|>.WN > 599.5
<|POSS_CA_PEAKS|>.WN < 600.5
=>Let CA_LINE_1.EXIST TRUE
=>Do <|POSS_CA_PEAKS|>.WN CA_LINE_1.WN
CARBON_LINE_1
CARBON_LINE
Yes OXYGEN_PRESENT
<|POSS_OX_PEAKS|>.WN > 1135.0
<|POSS_OX_PEAKS|>.WN < 1137.0
=>Let OX_LINE_1.EXIST TRUE
=>Do <|POSS_OX_PEAKS|>.WN OX_LINE_1.WN
OXYGEN_LINE_1
OXYGEN_LINE

Figure 17. The carbon and oxygen rules.

4.0 SAMPLE RUN

Table 3 is an example of the report that is produced by the expert system. For this sample run, it was given a silicon spectrum that contained the impurities boron, gallium, and oxygen. As seen in the report, the expert system was able to identify these impurities, determine their concentrations, and list their lines that were present.

Table 3. The report generated by the expert system.

```
              RESULTS OF IMPURITY CLASSIFICATION

                           11/13/1990

SPECTRUM NAME: siga819ca
TYPE: abs
-----------------------------------

        IMPURITY:  BORON
CONCENTRATION:   2.72301772882080081e15

           PEAKS DETECTED
***********************************
   LINE                WAVENUMBER
***********************************
     1                    244.96
     2                    278.23
    2p                   669.0
     3                   309.5
     4                    320.08
     8                    346.33
-----------------------------------

        IMPURITY:  GALLIUM
CONCENTRATION:   3.8677543243408203e16

           PEAKS DETECTED
***********************************
   LINE                WAVENUMBER
***********************************
     1                    469.85
    2p                    897.04
     3                    541.96
    3p                    921.13
     4                    550.03
     6                    569.19
     8                    574.21
    10                    582.28
    X1                    346.33
-----------------------------------

        IMPURITY:  OXYGEN
CONCENTRATION:   2.581750259399414050e16

           PEAKS DETECTED
***********************************
   LINE                WAVENUMBER
***********************************
     1                    1136.92
```

5.0 TESTING THE EXPERT SYSTEM

In deciding what should be considered in testing the expert system, four main criteria were settled upon:

1. Easy to use: The system should be easy to use with a minimum of experience needed in computer use.

2. Requires little analysis experience: The expert system should produce satisfactory results when being used by a lower level researcher such as a technician, co-op student or entry-level researcher.

3. Produces accurate results: The results generated by the expert system should agree with those of an experienced spectroscopist. These results are often quite subjective and vary considerably when the analysis is performed by different researchers. The expert system should find all the impurities present in a given sample spectra and the calculated impurity concentrations should be within the experimental tolerances that are considered acceptable in this type of spectral analysis.

4. Produces consistent results: The expert system should give consistent results if given identical spectra. This may seem like an obvious requirement, but the user is allowed some interaction with the system and can add peaks to the peak table if he feels they are significant when viewing the spectrum.

5.1 DISCUSSION

The expert system was first tested by an experienced spectroscopist with little computer experience. No significant difficulties were experienced by this user. After one tutorial session, the researcher was confident enough to run the system without assistance. A co-op student was the next test subject and required only one short tutorial session to become confident enough to use the system without assistance. The only interaction required by the expert system is for the user to s' pply a spectrum name and indicate whether it is an absorption coefficient or photoconductivity spectrum. Then the user is asked to examine the spectrum on the screen. This spectrum has all peaks marked that the expert system finds. The user may mark additional peaks if he feels they are significant. If these peaks are simply noise or some other irregularity in the spectrum, then the expert system ignores them. This is all the user is required to input to the system.

In order to test the accuracy of the expert system, its results were compared with the results of a study conducted several years earlier. This study examined the relationship between the area under a specific gallium peak in the spectrum and the gallium concentration for that sample as determined by other means such as Hall Analysis or Photoluminesence. The researcher conducting this study ran a series of gallium doped samples and carefully measured what he felt the gallium dopant-concentration peak areas were. The results of this study were used to calculate a concentration coefficient relating a specific gallium peak's area to the gallium concentration present in the sample.

Fourteen spectra were analyzed by the expert system in this test. Some runs were made by us, the designers, others by a co-op student working in the lab. The results are displayed in table 4.

Table 4. Results of gallium concentration study test.

SAMPLE ID	**GALLIUM CONCENTRATION ERROR (%)**
GA0102AC	-4.33
GA044BAC	-5.66
GA1004AC	-3.23
GA1069BAC	-1.46
GA1305AAC	-7.0
GA203AC	+0.6
GA370AC	+2.1
GA402AC	-6.6
GA757DAC	+ 0.1
GA757EAC	-2.5
GA757FAC	+0.42
GA987AC	+1.25
GA987BAC	-8.96
GA1072AC	-5.26

The expert system's results for impurity or dopant concentration averaged 2.9% less than the values calculated by the spectroscopist in this study. This figure is well within a reasonable value if comparing the results of two experienced researchers. Typically an error of 20% is not unreasonable due to differences that occur when determining the area of the calibration peak. A calibration peak is a peak occurring in a spectrum that is always generated by the presence of a particular dopant or impurity. The area of this peak is multiplied by a scale factor to calculate that particular dopant's concentration. If the individual chooses different peak boundaries, then different impurity or dopant concentrations will result. In a noisy spectrum, peak boundaries can be difficult to discern. This situation also occurs when the concentrations are very low and the peak is relatively weak. This can be seen in Figure 18; a typical weak peak occurring in a noisy area of the spectrum. Defining the limits of a peak in a noisy area can be difficult at best. This is where the differences generally occur when two researchers analyze the same spectrum.

The dopant or impurity peaks were classified correctly in all cases. The expert system always detected the presence of all impurities known to be present in the samples. In summary, there was good agreement between the system's analysis and those of other experienced researchers.

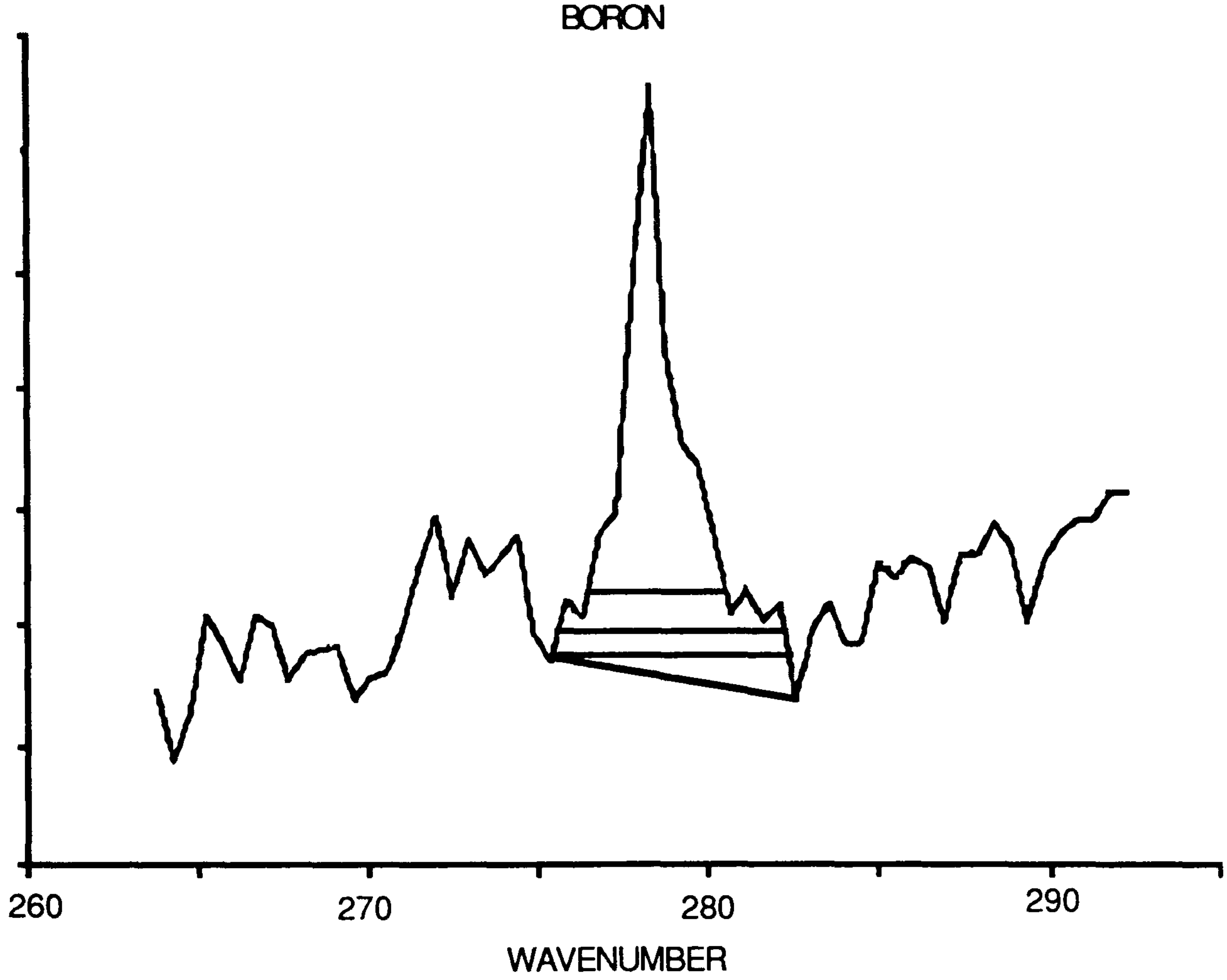


Figure 18. Possible peak boundaries in a noisy spectrum.

6.0 FUTURE PLANS FOR THE EXPERT SYSTEM

The expert-system is completed for use with high purity silicon materials. As stated in previous sections, the scope of the project was limited to silicon based electronic materials. The body of knowledge for these materials was well established. To have included other types of optical materials in the knowledge base would have required more man-hours of development and testing than could be justified. The main goal was to use this as a learning experience for future expert-system work within our laboratory. The completed system is dependable and could be used on a day-to-day basis for research as well as quality-control applications. Unfortunately, silicon based electronic materials are no longer the current materials of interest, so it is unlikely the system will be heavily used.

The next phase of this project is to design a knowledge base for analyzing superlattice materials grown by Molecular Beam Epitaxy (MBE). There currently is not a body of knowledge to draw upon for constructing this new system. When a sufficient number of samples of this type have been analyzed, a set of rules should emerge which will be the basis for the next phase. There has been discussion of designing a closed-loop system where the expert system can control the spectrometer as well as the sample chamber temperature, but this plan is still tentative and requires much more study as well as the full support of the researchers in this area.

APPENDIX A. FORTRAN SUBROUTINES

The expert-system calls several FORTRAN subroutines to perform routine analysis tasks. These routines are described below .

FILLSPECTRA The spectral array of data points is stored in a named Fortran common block. FILLSPECTRA is passed the name of the spectra to be analyzed. It simply locates the the spectra data file, opens it and loads the common block with the data points from this file. These data points are subsequently used for the interractive graphics in PEAKPICK and in calculations and graphics in FINDAREA.

PEAKPICK This is an interractive graphics routine which plots the subject spectrum on the workstation screen. All peaks that have been located by the FTIR data station are marked and the user is then asked if he wishes to mark additional peaks of interest. By using the attached mouse and cursor all peaks of interest that were not marked by the program can be marked by the user interractively. When complete this routine transfers to a subroutine called REFORM which creates an accompanying table of peaks from the spectrum under scrutiny. This table is in a format readable by the expert-system program. When the knowledge base is loaded this peak table is used as a basis for generating preliminary conclusions about the sample.

FINDAREA The impurity concentration is calculated by multiplying the area of a certain peak associated with that impurity by a calibration factor. The resulting impurity concentration is in the units of atoms/cm**2. Since these peaks are superimposed on a background signal that is often noisy, or the peak itself is very weak, defining the peak limits can be a subjective process. Different researchers can analyze the same spectra and obtain peak areas differing by 20%. This routine, after having been passed the wavenumber of an impurity peak of interest, displays suggested peak boundaries and asks the user if he agrees: if yes then calculations continue based on these peak limits. If no, the user indicates with the mouse and cursor what the peak boundaries should be and these are used for subsequent calculations. The peak area is then calculated by a trapezoidal integration method. The user interraction is handled by a subroutine called PEAKPLOT, called from FINDAREA. The calculated value of the area of the peak of interest is then passed back to the expert-system to be recorded and used in impurity concentration calculations. All FORTRAN callable screen graphics are produced by a graphics software package called CA-DISSPLA, distributed by Computer Associates Inc.

FILLSPECTRA

```
C ---------------------------------------------------------------
C
C  This routine reads in the spectrum points and loads them into
C  a common block called BLOCK1.
C
C ---------------------------------------------------------------

      INTEGER FUNCTION FILLSPECTRA(THESTR,NATOMS,THEATOMS)
       INCLUDE 'NXP$LIBRARY:NXPVMSDEF.FOR'
        CHARACTER*(*) THESTR
         INTEGER*4 THEATOMS(*)
        INTEGER*4 RETVAL
          LOGICAL FLAG  ! if returned TRUE then no errors
          INTEGER*4 NATOMS
          CHARACTER ANS
          CHARACTER*80 LINE
          CHARACTER*10 INFILE        !The datafile with no extension
          CHARACTER*15 DATAFILE      ! the spectra file

          COMMON /BLOCK1/X(20000), Y(20000),NUMBPTS,INFILE,FLAG
C         ! the spectral array of X-Y pairs

C
        RETVAL=NXP$GETATOMINFO(THEATOMS(1),NXP$_AINFO_VALUE,0,0,
          1      ,INFILE,NUMB1 )         ! the spectra file name

        IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
C          open the file to be examined
C
C         This section reads in the spectral array
C

c
          DATAFILE = INFILE//'.MAS'
          OPEN (FILE=DATAFILE,STATUS='OLD',UNIT=1,CARRIAGECONTROL='LIST',
          1  ERR=5000)

C          the following section reads the input file's header section
C          if there is one
C
            READ(1,30) LINE
3 0         FORMAT(A80)

            DO WHILE (LINE(1:2).EQ.'##')
             READ(1,30) LINE
           END DO

C         get the actual data X-Y pairs
           I=1
```

```
80          READ(1,*,END=95) X(I),Y(I)
          I=I+1
           NUMBPTS = I
          GO TO 80     ! read until an EOF is found
95       CLOSE( UNIT=1 )       ! normal exit point
         FILLSPECTRA = 1       ! return success
         RETURN

5000     print  5001
5001     format(' BAD FILE NAME ')
         return
         END
```

PEAKPICK

```
      INTEGER FUNCTION PEAKPICK()
C     ------------------------------------------------------------
C
C  This routine allows the user to view the spectrum with the
C   peaks marked. The user can then interractively mark add-
C   itional  peaks that he feels should be included in the
C   analysis of the sample.
C
C     ------------------------------------------------------------
C*******************       Variable    Declaration       ******************
C      X,Y            The x & y values of the whole spectra
C      Numbpts              The number of points in the spectra
C      Num            1/4 of the number of peaks in the whole spectra
C      Npk1-4               The number of points in each corresponding plot
C      X1,Y1          The temp var with the points to be plotted
C                       for each of the four plots
C      Peak1-4              The peaks picked by the user for each plot
C      Oldpkx,OldpkY  The peaks found in the .pk file
C      Name           The raw name of the spectra (no exts)
C      Line           Dummy variable used to read in a header line
      INCLUDE 'NXP$LIBRARY:NXPVMSDEF.FOR'
      Common /Block1/X(20000), y(20000), Numbpts, Name,flag

      Integer              Num,Npk1,Npk2,Npk3,Npk4,numold
      real           X1(5000),  Y1(5000)
      real           Peak1(25,2),Peak2(25,2),Peak3(25,2),Peak4(25,2)
      real           OldPkx(100),  OldPky(100)
      character*10         Name
      character*15         infile
      character*80         Line
      logical        flag, exist

C*********************          Begin     Code          ***********************

      IF  (EXIST(name//'.pk'))  THEN
            Open(unit=1,status='old',file=name//'.pk',err=5000)
            ! Loop to get rid of header in .pk file
            do 20 I=1,6
               read(1,10) line
10             format(a80)
20          continue

            ! Loop to read in the old peaks from the .pk file
            I=1
30          read(1,*,end=45)  OldPkx(I),  OldPky(I)
               I= I + 1
               go to 30
40          continue
45          numold = I
      else
```

```
                open(unit=1,status='new',file=name//'.pk')
             do 50 I=1,6
                 write(1,*) 'Header Line'
50              continue
                Numold = 0
        endif
C**     Here we send the plot to the screen in four different parts, having
C**      the user return the peaks he sees on each of the four plots


          Num = Numbpts/4

C******        First Plot
          K=0
          Do 60 I=1,Num
             K=K+1
               x1(k)=x(I)
              Y1(k)=Y(I)
60        continue
          call  Plot(X1,Y1,K,Name,Peak1,Npk1,oldpkx,oldpky,numold)

C******        Second Plot
          K=0
          Do  70  I=(Num-100),(Num*2)
             K=K+1
               x1(k)=x(I)
              Y1(k)=Y(I)
70        continue
          call  Plot(X1,Y1,K,Name,Peak2,Npk2,oldpkx,oldpky,numold)

C******         Third Plot
          K=0
          Do  80  I=(Num*2-100),(Num*3)
             K=K+1
               x1(k)=x(I)
              Y1(k)=Y(I)
80        continue
          call  Plot(X1,Y1,K,Name,Peak3,Npk3,oldpkx,oldpky,numold)

C******         Fourth Plot
          K=0
          Do  90  I=(Num*3-100),Numbpts
             K=K+1
               x1(k)=x(I)
              Y1(k)=Y(I)
90        continue
          call  Plot(X1,Y1,K,Name,Peak4,Npk4,oldpkx,oldpky,numold)

          call Donepl    !finished  plotting
          CALL SETDEV(IERR,0)
          close(1)       !close the .pk file
```

```
C****  Now we take the four peak arrays and append them to the Name.pk
C****   file that already exist

        infile = name//'.pk'
        IF (EXIST(infile)) then
             Open(unit=10,status='old', access='append',file=infile)
        ELSE
             Open(unit=10,status='new', file=infile)
        END IF

        do 500 I=1,Npk1
           Write(10,*) peak1(I,1), peak1(I,2)
500     Continue

        do 510 I=1,Npk2
           Write(10,*) peak2(I,1), peak2(I,2)
510     Continue

        do 520 I=1,Npk3
           Write(10,*) peak3(I,1), peak3(I,2)
520     Continue

        do 530 I=1,Npk4
           Write(10,*) peak4(I,1), peak4(I,2)
530     Continue

        close(10)     !close the .pk file
        PEAKPICK = 1        ! return success code
        CALL REFORM         ! make a new .nxp peak table
        return
c
c
5000      write(*,*) name//'.pk', 'error opening that file!!'
        FLAG = .FALSE.
        call exit
        end

C******************     PLOT     *****************************
        subroutine Plot(X,Y,N,Name,Peak,NumPts,oldpkx,oldpky,numold)

        real          X(5000),  Y(5000),  Peak(100),olpkx(100),oldpky(100)
        real          Xmin,Xstep,Xmax,Ymin,Ystep,Ymax
        Integer             N, NumPts, NumOld
        Character*10        Name

        N=N-1
      call PGPX
        call newclr('whit')
       call hwspec(0.0, 'BACK')
       call page(13,11.)
        call area2d(12,10)
       call NEWCLR('viol')
       call  Headin(name,100,1.5,1)
       call  Rndlin(X,N,11.5,Xmin,Xstep,Xmax)
       call  Rndlin(Y,N,9.5,Ymin,Ystep,Ymax)
       call  Xname('Wave  Number',100)
       call  Yname('Range',100)
       call  Graf(Xmin,Xstep,Xmax,Ymin,Ystep,Ymax)
       call  Newclr('red')
       call  Curve(X,Y,N,0)
       call  Newclr('Gree')
       call   Marker(13)
       call  Setdev(0,0)
       call  Curve(oldpkx,oldpky,Numold,-1)
       Call  messag('Select peaks with left mouse button.',37,1.0,9.75)
       Call  Messag('Click  middle  button  to  continue.',31,7.5,9.75)
       call  Read_Screen(Xmin,Xmax,Ymin,Ymax,Peak,Numpts)
       call  Endpl(0)
       return
       end

C******************************      Subroutine    Exist        ****************

     Function Exist(FileName)
C
C Tests for the existance of a filename.
C
     Character*15 FileName
    Integer IERR
    Logical Exist
C
C Test for file at CurDate
C
     OPEN (File = FileName,status = 'OLD',Unit = 1,iostat = ierr)
    If (IERR.EQ.29) then
     Exist = .FALSE.
   else
     Exist = .TRUE.
   end if


C
     Close(Unit = 1)
    Return
   End

C******************            Read_screen
************************************

       subroutine  read_screen  (xmin,xmax,ymin,ymax,Peak,NumPts)

       real    xpos            !x position read from the screen
       real    ypos            !y position read from the screen
       integer  i                !number of points read form the screen
       real    save_xpos(100)  !the numbers inputed from the screen
```

```
      real    save_ypos(100)  !the numbers inputed from the screen
      logical        first          !is this the first pos read from the screen?
      integer        rkey           !status of the return key (0=o.k., 1=stop)
      real   Peak(25,2)
      Integer NumPts

C  Use the mouse to get the positions from the screen
C       call  angle(9.0)
      first = .true.
      rkey = 0
      I=0
      do while (rkey .ne. 1)
10      continue
          call  get_position(xmin,xmax,ymin,ymax,first,
   &                   xpos,ypos,rkey)
        if (rkey .eq. 0) then
            call  MARKER(13)
           call CURVE(Xpos,Ypos,1,-1)
            I = I + 1
            Peak(I,1) = Xpos
            Peak(I,2) = Ypos
            save_xpos(I) = xpos
            save_ypos(I) = ypos
            go to 10
        else if (rkey .eq. 1) then
            NumPts = I
            return
              else
                  type 300
300               format(' No valid data available')
        endif
      enddo
      call ANGLE(0.0)
      return
      end

C******************          get_position          ****************************

      subroutine GET_POSITION(xmin,xmax,ymin,ymax,
   &                     first,xpos,ypos,rkey)

      Integer              rkey   !status return key (0=ok,1=stop)
      Real          xmin,xmax,ymin,ymax
      Real          pos, Ypos       !returned values from the plot
      Logical              First
      Integer              IKEY_R
      Real          Initx,  Inity    !initial cross-hair position


      if  (first)  then
        call CRDSYS('USER')
          initx=(xmax-xmin)/2.0
      inity=(ymax-ymin)/2.0
    firsts = .false.
  endif
  call  INILOC(1,xpos,ypos,0,0)
  call  ECHLOC(1,2,xmin,xmax,ymin,ymax)
  call  REQLOC(1,rkey,xpos,ypos,ikey_r)

  return
  end
```

FINDAREA

```
C          -----------------------------------------------------------
C
C  This routine is passed the left and right x-coordinates of a peak in a spectral
C   data file. This subroutine calculates the area of that peak and returns it to
C   the expert system for later use.
C
C          -----------------------------------------------------------

     SUBROUTINE FINDAREA(THESTR,NATOMS,THEATOMS)
     INCLUDE 'NXP$LIBRARY:NXPVMSDEF.FOR'
      CHARACTER*(*) THESTR
      INTEGER*4 THEATOMS(*),STAT
     INTEGER*4 RETVAL,RETVAL1,RETVAL2,RETVAL3,IDFLAG
         LOGICAL FLAG  ! if returned TRUE then no errors
         INTEGER*4 NATOMS

         COMMON /BLOCK1/X(20000), Y(20000),LOADED,NUMBPTS
C        ! the spectral array of X-Y pairs

         CHARACTER*80 LINE
         CHARACTER*15 INFILE        ! the spectra file
         CHARACTER*15 INCODE
         CHARACTER*2 CODE
         REAL LASTX,MR,ML,L_EDGE,PEAK
         INTEGER DELTA,RINDEX

C
      RETVAL1=NXP$GETATOMINFO(THEATOMS(1),NXP$_AINFO_VALUE,0,0,
     $  ,INFILE,NUMB1 )         ! the spectra file name
C
      RETVAL2=NXP$GETATOMINFO(THEATOMS(2),NXP$_AINFO_VALUE,0,0,
    $  NXP$_DESC_FLOAT,PEAK,NUMB2 )   ! the peak location
C
      RETVAL3=NXP$GETATOMINFO(THEATOMS(3),NXP$_AINFO_VALUE,0,0,
     $  ,INCODE,NUMB3 )         ! the type of dopant code

      IF (RETVAL1 .NE. NXP$_ERR_NOERR) GO TO 5000
      IF (RETVAL2 .NE. NXP$_ERR_NOERR) GO TO 5000
      IF (RETVAL3 .NE. NXP$_ERR_NOERR) GO TO 5000
C
         CODE= INCODE(1:NUMB3)    ! cut off excess stuff
C
C
         IPEAKX=IFIND(PEAK)         ! get set up for loop
         IRIGHT= 0
         LEFT= 10
         MR  =  -20
         ML=  20

C  This loop works its way down both sides of the peak, computing the
```

```
C  average slope on either side. When one side bottoms out ( slope is
C  zero ) it stops.

      DO WHILE( (ML.GT.0) .AND. (MR.LT.0))
        MR= SLOPE(IPEAKX+IRIGHT)        ! get the right side slope
        ML= SLOPE(IPEAKX-LEFT)          ! get the left side slope
        LEFT= LEFT+1                    ! farther down left
        IRIGHT= IRIGHT+1  ! farther down right
      END DO

      IF (ML.LE.0) DELTA= LEFT + 6       ! why 6 and 4, you'll have to
      IF (MR.GE.0) DELTA= IRIGHT + 4   ! draw a picture and convince
                                        ! yourself that these work

      LINDEX= IPEAKX - DELTA       ! the left edge index of the peak
      RINDEX= IPEAKX + DELTA       ! the right edge index
      L_EDGE = X(LINDEX)
      R_EDGE = X(RINDEX)
C
      CALL PEAKPLOT(L_EDGE,R_EDGE,CODE)     ! let the user look at the peak
C                                   ! limits & change them if needed
C
      CALL PKAREA(L_EDGE,R_EDGE,AREA)       ! get the area
C
      IF ( AREA .LE. 0) GO TO 5000           ! if its a bad area, say so
C
C
      IF (CODE.EQ.'AL') THEN           !  if it's an aluminum peak
       RETVAL=NXP$GETATOMID('ALUMINUM_CALIBRATION_PEAK.AREA',
      1 IDFLAG,NXP$_ATYPE_SLOT)        ! get the area address
     IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
C      RETVAL=NXP$GETATOMID('ALUMINUM_CALIBRATION_PEAK.START',
C     1 IDFLAG1,NXP$_ATYPE_SLOT)       ! get the l-edge address
C     IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
C      RETVAL=NXP$GETATOMID('ALUMINUM_CALIBRATION_PEAK.END',
C     1 IDFLAG2,NXP$_ATYPE_SLOT)       ! get the R-edge address
C     IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
c     volunteer the area value
       RETVAL=NXP$VOLUNTEER(IDFLAG,NXP$_DESC_FLOAT,AREA,
      1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
     IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
c     volunteer the l-edge value
C      RETVAL=NXP$VOLUNTEER(IDFLAG1,NXP$_DESC_FLOAT,L_EDGE,
C     1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C     IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
C     volunteer the r-edge value
C      RETVAL=NXP$VOLUNTEER(IDFLAG2,NXP$_DESC_FLOAT,R_EDGE,
C       1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
         GO TO 95
        ENDIF
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
        IF (CODE.EQ.'CA') THEN      ! if it's an carbon peak
         RETVAL=NXP$GETATOMID('CARBON_CALIBRATION_PEAK.AREA',
        1 IDFLAG,NXP$_ATYPE_SLOT)        ! get the area address
      IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
C        RETVAL=NXP$GETATOMID('CARBON_CALIBRATION_PEAK.START',
C       1 IDFLAG1,NXP$_ATYPE_SLOT)       ! get the l-edge address
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
C        RETVAL=NXP$GETATOMID('CARBON_CALIBRATION_PEAK.END',
C       1 IDFLAG2,NXP$_ATYPE_SLOT)       ! get the R-edge address
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
c       volunteer the area value
         RETVAL=NXP$VOLUNTEER(IDFLAG,NXP$_DESC_FLOAT,AREA,
        1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
      IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
c       volunteer the l-edge value
C        RETVAL=NXP$VOLUNTEER(IDFLAG1,NXP$_DESC_FLOAT,L_EDGE,
C       1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
C
C       volunteer the r-edge value
C        RETVAL=NXP$VOLUNTEER(IDFLAG2,NXP$_DESC_FLOAT,R_EDGE,
C       1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
         GO TO 95
        ENDIF
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        IF (CODE.EQ.'OX') THEN      ! if it's an oxygen peak
         RETVAL=NXP$GETATOMID('OXYGEN_CALIBRATION_PEAK.AREA',
        1 IDFLAG,NXP$_ATYPE_SLOT)        ! get the area address
      IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C        RETVAL=NXP$GETATOMID('OXYGEN_CALIBRATION_PEAK.START',
C       1 IDFLAG1,NXP$_ATYPE_SLOT)       ! get the l-edge address
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C        RETVAL=NXP$GETATOMID('OXYGEN_CALIBRATION_PEAK.END',
C       1 IDFLAG2,NXP$_ATYPE_SLOT)       ! get the R-edge address
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

c       volunteer the area value
         RETVAL=NXP$VOLUNTEER(IDFLAG,NXP$_DESC_FLOAT,AREA,
        1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

c       volunteer the l-edge value
C         RETVAL=NXP$VOLUNTEER(IDFLAG1,NXP$_DESC_FLOAT,L_EDGE,
C       1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C       volunteer the r-edge value
C         RETVAL=NXP$VOLUNTEER(IDFLAG2,NXP$_DESC_FLOAT,R_EDGE,
C       1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
          GO TO 95
        ENDIF
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        IF (CODE.EQ.'BO') THEN        ! if it's a boron peak

C       get the area address
          RETVAL=NXP$GETATOMID('BORON_CALIBRATION_PEAK.AREA',
        1 IDFLAG,NXP$_ATYPE_SLOT)
       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C         RETVAL=NXP$GETATOMID('BORON_CALIBRATION_PEAK.START',
C       1 IDFLAG1,NXP$_ATYPE_SLOT)       ! get the l-edge address
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C         RETVAL=NXP$GETATOMID('BORON_CALIBRATION_PEAK.END',
C       1 IDFLAG2,NXP$_ATYPE_SLOT)       ! get the R-edge address
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C       volunteer the area value
          RETVAL=NXP$VOLUNTEER(IDFLAG,NXP$_DESC_FLOAT,AREA,
        1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

c       volunteer the l-edge value
C         RETVAL=NXP$VOLUNTEER(IDFLAG1,NXP$_DESC_FLOAT,L_EDGE,
C       1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C       volunteer the r-edge value
C         RETVAL=NXP$VOLUNTEER(IDFLAG2,NXP$_DESC_FLOAT,R_EDGE,
C       1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C       IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
          GO TO 95
        ENDIF
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
        IF (CODE.EQ.'IN') THEN        ! if it's an indium peak

c       get the area address
```

```
           RETVAL=NXP$GETATOMID('INDIUM_CALIBRATION_PEAK.AREA',
          1 IDFLAG,NXP$_ATYPE_SLOT)
        IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C          RETVAL=NXP$GETATOMID('INDIUM_CALIBRATION_PEAK.START',
C         1 IDFLAG1,NXP$_ATYPE_SLOT)      ! get the l-edge address
C         IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C          RETVAL=NXP$GETATOMID('INDIUM_CALIBRATION_PEAK.END',
C         1 IDFLAG2,NXP$_ATYPE_SLOT)      ! get the R-edge address
C         IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

           RETVAL=NXP$VOLUNTEER(IDFLAG,NXP$_DESC_FLOAT,AREA,
          1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
        IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

c         volunteer the l-edge value
C          RETVAL=NXP$VOLUNTEER(IDFLAG1,NXP$_DESC_FLOAT,L_EDGE,
C         1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C         IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C         volunteer the r-edge value
C          RETVAL=NXP$VOLUNTEER(IDFLAG2,NXP$_DESC_FLOAT,R_EDGE,
C         1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C         IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000
           GO TO 95
          ENDIF
C
C-------------------------------------------------------------------------
          IF (CODE.EQ.'GA') THEN        ! if it's a gallium peak

C         get the area address
           RETVAL=NXP$GETATOMID('GALLIUM_CALIBRATION_PEAK.AREA',
          1 IDFLAG,NXP$_ATYPE_SLOT)
        IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C          RETVAL=NXP$GETATOMID('GALLIUM_CALIBRATION_PEAK.START',
C         1 IDFLAG1,NXP$_ATYPE_SLOT)      ! get the l-edge address
C         IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C          RETVAL=NXP$GETATOMID('GALLIUM_CALIBRATION_PEAK.END',
C         1 IDFLAG2,NXP$_ATYPE_SLOT)      ! get the R-edge address
C         IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C         volunteer the area value
           RETVAL=NXP$VOLUNTEER(IDFLAG,NXP$_DESC_FLOAT,AREA,
          1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
        IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

c         volunteer the l-edge value
C          RETVAL=NXP$VOLUNTEER(IDFLAG1,NXP$_DESC_FLOAT,L_EDGE,
C         1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
```

```
C     IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

C     volunteer the r-edge value
C       RETVAL=NXP$VOLUNTEER(IDFLAG2,NXP$_DESC_FLOAT,R_EDGE,
C     1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))
C     IF (RETVAL .NE. NXP$_ERR_NOERR) GO TO 5000

       GO TO 95
      ENDIF

5000  FLAG = .FALSE.          ! only exit here if an error occurred
      RETVAL=NXP$GETATOMID('STATUS.FINDAREA',IDFLAG,NXP$_ATYPE_SLOT)
      RETVAL=NXP$VOLUNTEER(IDFLAG,NXP$_DESC_INT,FLAG,
      1 JIOR(NXP$_VSTRAT_NOFWRD,NXP$M_VSTRAT_SET))

95    CLOSE( UNIT=1 )      ! normal exit point
96    RETURN
      END
C      ----------------------------------------------------------------
C
       FUNCTION IFIND(POINT)
C This function returns the x index value of POINT in SPECTRA

      COMMON /BLOCK1/X(20000), Y(20000),LOADED,NUMBPTS
C
      I=1
      DO WHILE (X(I).LT.POINT)
            I=I+1
      END DO
      IFIND=I
      RETURN
      END

C -------------------------------------------------------------------
C
      FUNCTION SLOPE(J)
C
C This routine calculates the average slope of a curve segment. It
C currently uses 10 points to average. The parameter J is the index
C number of the leftmost point in the slope averaging 'window'.
C

      COMMON /BLOCK1/X(20000), Y(20000),LOADED,NUMBPTS
      REAL M,NUM
C
      M=0.0
      DO 10 I=1 , 10
         NUM= Y(J+1)- Y(J)
         DENOM= X(J+1)- X(J)
       IF(DENOM.EQ.0) DENOM=OLDENOM
       M= NUM/DENOM + M
       OLDENOM= DENOM
```

```
        J= J + 1
10     CONTINUE
       SLOPE= M/10.0
       RETURN
       END

C-------------------------------------------------------------
       Subroutine PKAREA(STP, ENP, AREA)
C
C
C      This program, given a set of endpoints and the (x,y) points
C      in between, will calculate the area under the peak using
C      a trapezoidal approximation.  First calculating the area
C      from the peak to thge zero baseline and then subtracting
C      from that the area from the zero baseline to the peak baseline.
C
C      Variable Declarations
C      -------------------
C      STP      -> starting point of the peak
C      ENP      -> ending point of the peak
C      SPECTRA    -> all the points in the spectra represented by
C                     their (x,y) values in the array form:
C                   (data #, x or y) = array representation
C                              (1,x)  =  (1,1)
C                              (1,y)  =  (1,2)
C                              (2,x)  =  (2,1)
C                              (2,y)  =  (2,2)
C                                  :
C                                  :
C                              (n,x)  =  (n,1)
C                              (n,y)  =  (n,2)
C      RESAR    -> area from peak baseline to zero baseline
C      SUM      -> area under the peak and above the zero baseline
C      AREA     -> the area under the peak and above the peak baseline
C      I        -> loop control variables

       Common /BLOCK1/X(20000), Y(20000),LOADED,NUMBPTS
       Real EP(2), SP(2)

C      **********************************************************
C      *    Loop to find the starting point in the spectra       *
C      **********************************************************

       I = 1

       Do While (STP .gt. X(I))
             I = I + 1
       End Do
       I = I - 1                  I back up a hair
       SP(1) = X(I)
       SP(2) = Y(I)
```

```
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       *Loop to calculate the area from the peak to the zero baseline *
C                      ***************************************************************

        SUM = 0.0

        Do While (ENP .ge. X(I))
               SUM = SUM + 0.5 * (X(I+1) - X(I)) *
     &  (Y(I) + Y(I+1))
               I = I + 1

        Enddo
        EP(1) = X(I)
        EP(2) = Y(I)

C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       *      Calculate RESAR and subtract it from SUM          *
C       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

        RESAR = 0.5 * (EP(1) - SP(1))*(EP(2) + SP(2))
        AREA = SUM - RESAR
        RETURN
        END
```

APPENDIX B. LIST OF RULES

```
RULE : Rule INVESTIGATE_ALUMINUM_ABS (#1)
If
    there is evidence of ALUMINUM_PRESENT
    And SPECTRA.TYPE is "ABS"
Then ALUMINUM_ANALYZED_ABS
    is confirmed.
    And CODE is set to "AL"
    And Execute
"FINDAREA"(@ATOMID=SPECTRA.POINTS,ALUMINUM_CALIBRATION_
PEAK.WAVENUMBER,CODE.Value;)
    And (ALUMINUM_CALIBRATION_PEAK.AREA*4.1666e+13) is
        assigned to ALUMINUM_CALIBRATION_PEAK.CONC

RULE : Rule INVESTIGATE_ALUMINUM_PHOTO (#2)
If
    there is evidence of ALUMINUM_PRESENT
    And SPECTRA.TYPE is "PHOTO"
Then ALUMINUM_ANALYZED_PHOTO
    is confirmed.
    And CODE is set to "AL"
    And 0.0 is assigned to ALUMINUM_CALIBRATION_PEAK.AREA

RULE : Rule ALUMINUM_LINE_X4 (#3)
If
    there is evidence of ALUMINUM_PRESENT
    And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
        than 413.5
    And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
        414.5
Then ALUMINUM_LINE
    is confirmed.
    And AL_LINE_X4.EXIST is set to TRUE
    And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
        to AL_LINE_X4.WN
```

```
RULE : Rule ALUMINUM_LINE_X3 (#4)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 403.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                404.5
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_X3.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_X3.WN

RULE : Rule ALUMINUM_LINE_X2 (#5)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 371.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                372.5
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_X2.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_X2.WN

RULE : Rule ALUMINUM_LINE_X1 (#6)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 337.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                338.5
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_X1.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_X1.WN
```

```
RULE :  Rule ALUMINUM_LINE_EX (#7)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  453.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                4454.5
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_EX.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_EX.WN

RULE :  Rule ALUMINUM_LINE_ESTAR (#8)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  911.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                912
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_ESTAR.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_ESTAR.WN

RULE :  Rule ALUMINUM_LINE_EI (#9)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  556.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                557
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_EI.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_EI.WN
```

```
RULE : Rule ALUMINUM_LINE_9 (#10)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 546
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                548
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_9.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_9.WN

RULE : Rule ALUMINUM_LINE_8 (#11)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 543
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                546
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_8.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_8.WN

RULE : Rule ALUMINUM_LINE_7 (#12)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 540.7
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                542
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_7.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_7.WN
```

```
RULE : Rule ALUMINUM_LINE_6 (#13)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 538
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                539.3
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_6.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_6.WN

RULE : Rule ALUMINUM_LINE_5 (#14)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 534.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                535.5
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_5.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_5.WN

RULE : Rule ALUMINUM_LINE_4P (#15)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 900
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                900.5
Then ALUMINUM_LINE
        is confirmed.
        And AL_LINE_4P.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_4P.WN
```

```
RULE :  Rule ALUMINUM_LINE_3P (#16)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  891.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                892
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_3P.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_3P.WN

RULE :  Rule ALUMINUM_LINE_2P (#17)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  867
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                868
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_2P.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_2P.WN

RULE :  Rule ALUMINUM_LINE_2 (#18)
If
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  471
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                472.5
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_2.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_2.WN
```

```
RULE :  Rule ALUMINUM_LINE_11 (#19)
I f
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  554.5
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                556.4
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_11.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_11.WN

RULE :  Rule ALUMINUM_LINE_10 (#20)
I f
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  549
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                554
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_10.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_10.WN

RULE :  Rule ALUMINUM_LINE_1 (#21)
I f
        there is evidence of ALUMINUM_PRESENT
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than  442
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                443.5
Then ALUMINUM_LINE
        is  confirmed.
        And AL_LINE_1.EXIST is set to TRUE
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is assigned
                to AL_LINE_1.WN
```

```
RULE : Rule INVESTIGATE_ALUMINUM (#22)
If
        there is evidence of POSSIBLE_ALUMINUM
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is greater
                than 441.7
        And <|POSSIBLE_ALUMINUM_PEAKS|>.WAVENUMBER is less than
                443.7
        And <<|POSSIBLE_ALUMINUM_PEAKS|>>.WAVENUMBER is greater
                than 470.9
        And <<|POSSIBLE_ALUMINUM_PEAKS|>>.WAVENUMBER is less
                than 472.9
        And <<<|POSSIBLE_ALUMINUM_PEAKS|>>>.WAVENUMBER is
                greater than 542.9
        And <<<|POSSIBLE_ALUMINUM_PEAKS|>>>.WAVENUMBER is less
                than 544.9
Then ALUMINUM_PRESENT
        is confirmed.
        And <<|POSSIBLE_ALUMINUM_PEAKS|>>.WAVENUMBER is assigned
                to ALUMINUM_CALIBRATION_PEAK.WAVENUMBER

RULE : Rule START_THINGS_OFF (#23)
If
        there is evidence of PEAKS_LOADED
        And there is evidence of PEAKS_EXAMINED
        And there is evidence of PEAKS_ANALYZED
Then ANALYZED
        is confirmed.

RULE : Rule INVESTIGATE_BORON_ABS (#24)
If
        there is evidence of BORON_PRESENT
        And SPECTRA.TYPE is "ABS"
Then BORON_ANALYZED_ABS
        is confirmed.
        And CODE is set to "BO"
        And Execute
"FINDAREA"(@ATOMID=SPECTRA.POINTS,BORON_CALIBRATION_PEAK.
WAVENUMBER,CODE.Value;)
        And (BORON_CALIBRATION_PEAK.AREA*1.5151e+13) is
                assigned to BORON_CALIBRATION_PEAK.CONC
```

```
RULE : Rule INVESTIGATE_BORON_PHOTO (#25)
If
        there is evidence of BORON_PRESENT
        And SPECTRA.TYPE is "PHOTO"
Then BORON_ANALYZED_PHOTO
        is  confirmed.
        And CODE is set to "BO"
        And 0.0 is assigned to BORON_CALIBRATION_PEAK.AREA

RULE : Rule BORON_LINE_X4 (#26)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                260.0
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                262.0
Then BORON_LINE
        is  confirmed.
        And BO_LINE_X4.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_X4.WN

RULE : Rule BORON_LINE_X3 (#27)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                249.0
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                251.0
Then BORON_LINE
        is  confirmed.
        And BO_LINE_X3.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_X3.WN
```

```
RULE : Rule BORON_LINE_X2 (#28)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                219.0
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                221.0
Then BORON_LINE
        is confirmed.
        And BO_LINE_X2.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_X2.WN

RULE : Rule BORON_LINE_X1 (#29)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                183.5
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                184.5
Then BORON_LINE
        is confirmed.
        And BO_LINE_X1.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_X1.WN

RULE : Rule BORON_LINE_EX (#30)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                298.0
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                300.0
Then BORON_LINE
        is confirmed.
        And BO_LINE_EX.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_EX.WN
```

```
RULE : Rule BORON_LINE_ESTAR (#31)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                712.4
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                713.4
Then BORON_LINE
        is  confirmed.
        And BO_LINE_ESTAR.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_ESTAR.WN

RULE : Rule BORON_LINE_E (#32)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                358.0
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                359.0
Then BORON_LINE
        is  confirmed.
        And BO_LINE_E.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_E.WN

RULE : Rule BORON_LINE_9 (#33)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                348.6
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                349.3
Then BORON_LINE
        is  confirmed.
        And BO_LINE_9.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_9.WN
```

```
RULE :  Rule BORON_LINE_8 (#34)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                344.2
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                346.7
Then BORON_LINE
        is  confirmed.
        And BO_LINE_8.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_8.WN

RULE :  Rule BORON_LINE_7 (#35)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                341.6
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                342.6
Then BORON_LINE
        is  confirmed.
        And BO_LINE_7.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_7.WN

RULE :  Rule BORON_LINE_6 (#36)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                337.5
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                340.6
Then BORON_LINE
        is  confirmed.
        And BO_LINE_6.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_6.WN
```

```
RULE :  Rule BORON_LINE_5 (#37)
I f
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                334.0
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                335.0
Then BORON_LINE
        is  confirmed.
        And BO_LINE_5.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_5.WN

RULE :  Rule BORON_LINE_4P (#38)
I f
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                700.8
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                701.8
Then BORON_LINE
        is  confirmed.
        And BO_LINE_4P.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_4P.WN

RULE :  Rule BORON_LINE_4 (#39)
I f
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                318.8
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                322.4
Then BORON_LINE
        is  confirmed.
        And BO_LINE_4.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_4.WN
```

```
RULE :  Rule BORON_LINE_3P (#40)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                692.3
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                693.4
Then BORON_LINE
        is  confirmed.
        And BO_LINE_3P.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_3P.WN

RULE :  Rule BORON_LINE_3 (#41)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                308.8
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                309.8
Then BORON_LINE
        is  confirmed.
        And BO_LINE_3.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_3.WN

RULE :  Rule BORON_LINE_2P (#42)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                668.1
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                669.1
Then BORON_LINE
        is  confirmed.
        And BO_LINE_2P.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_2P.WN
```

```
RULE :  Rule BORON_LINE_2 (#43)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                277.7
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                278.7
Then BORON_LINE
        is  confirmed.
        And BO_LINE_2.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_2.WN

RULE :  Rule BORON_LINE_11 (#44)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                356.0
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                357.6
Then BORON_LINE
        is  confirmed.
        And BO_LINE_11.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_11.WN

RULE :  Rule BORON_LINE_10 (#45)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                350.3
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                356.0
Then BORON_LINE
        is  confirmed.
        And BO_LINE_10.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_10.WN
```

```
RULE : Rule BORON_LINE_1 (#46)
If
        there is evidence of BORON_PRESENT
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                244.5
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                245.5
Then BORON_LINE
        is confirmed.
        And BO_LINE_1.EXIST is set to TRUE
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is assigned to
                BO_LINE_1.WN

RULE : Rule INVESTIGATE_BORON (#47)
If
        there is evidence of POSSIBLE_BORON
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is greater than
                244.5
        And <|POSSIBLE_BORON_PEAKS|>.WAVENUMBER is less than
                245.5
        And <<|POSSIBLE_BORON_PEAKS|>>.WAVENUMBER is greater than
                277.2
        And <<|POSSIBLE_BORON_PEAKS|>>.WAVENUMBER is less than
                279.2
        And <<<|POSSIBLE_BORON_PEAKS|>>>.WAVENUMBER is greater
                than 318.3
        And <<<|POSSIBLE_BORON_PEAKS|>>>.WAVENUMBER is less than
                320.3
Then BORON_PRESENT
        is confirmed.
        And <<|POSSIBLE_BORON_PEAKS|>>.WAVENUMBER is assigned to
                BORON_CALIBRATION_PEAK.WAVENUMBER
```

```
RULE : Rule INVESTIGATE_CARBON_ABS (#48)
If
        there is evidence of CARBON_PRESENT
        And SPECTRA.TYPE is "ABS"
Then CARBON_ANALYZED_ABS
        is confirmed.
        And CODE is set to "CA"
        And Fxecute
"FINDAREA"(@ATOMID=SPECTRA.POINTS,CARBON_CALIBRATION_PEAK.
WAVENUMBER,CODE.Value;)
        And (CARBON_CALIBRATION_PEAK.AREA*6.700e+16) is
                assigned to CARBON_CALIBRATION_PEAK.CONC

RULE : Rule INVESTIGATE_CARBON_PHOTO (#49)
If
        there is evidence of CARBON_PRESENT
        And SPECTRA.TYPE is "PHOTO"
Then CARBON_ANALYZED_PHOTO
        is confirmed.
        And CODE is set to "CA"
        And 0.0 is assigned to CARBON_CALIBRATION_PEAK.AREA

RULE : Rule CARBON_LINE_1 (#50)
If
        there is evidence of CARBON_PRESENT
        And <|POSSIBLE_CARBON_PEAKS|>.WAVENUMBER is greater than
                599.5
        And <|POSSIBLE_CARBON_PEAKS|>.WAVENUMBER is less than
                600.5
Then CARBON_LINE
        is confirmed.
        And CA_LINE_1.EXIST is set to TRUE
        And <|POSSIBLE_CARBON_PEAKS|>.WAVENUMBER is assigned to
                CA_LINE_1.WN
```

```
RULE :  Rule INVESTIGATE_CARBON (#51)
If
        there is evidence of POSSIBLE_CARBON
        And <|POSSIBLE_CARBON_PEAKS|>.WAVENUMBER is greater than
                599.5
        And <|POSSIBLE_CARBON_PEAKS|>.WAVENUMBER is less than
                600.5
Then CARBON_PRESENT
        is  confirmed.
        And <|POSSIBLE_CARBON_PEAKS|>.WAVENUMBER is assigned to
                CARBON_CALIBRATION_PEAK.WAVENUMBER

RULE :  Rule AL_LINES (#52)
If
        there is evidence of ANALYZED
        And there is evidence of ALUMINUM_PRESENT
Then FIND_AL_LINES
        is  confirmed.
        And ALUMINUM_LINE is assigned to ALUMINUM_LINE

RULE :  Rule BO_LINES (#53)
If
        there is evidence of ANALYZED
        And there is evidence of BORON_PRESENT
Then FIND_BO_LINES
        is  confirmed.
        And BORON_LINE is assigned to BORON_LINE

RULE :  Rule CA_LINES (#54)
If
        there is evidence of ANALYZED
        And there is evidence of CARBON_PRESENT
Then FIND_CA_LINES
        is  confirmed.
        And CARBON_LINE is assigned to CARBON_LINE
```

```
RULE :  Rule GA_LINES (#55)
If
        there is evidence of ANALYZED
        And there is evidence of GALLIUM_PRESENT
Then FIND_GA_LINES
        is  confirmed.
        And GALLIUM_LINE is assigned to GALLIUM_LINE

RULE :  Rule IN_LINES (#56)
If
        there is evidence of ANALYZED
        And there is evidence of INDIUM_PRESENT
Then FIND_IN_LINES
        is  confirmed.
        And INDIUM_LINE is assigned to INDIUM_LINE

RULE :  Rule OX_LINES (#57)
If
        there is evidence of ANALYZED
        And there is evidence of OXYGEN_PRESENT
Then FIND_OX_LINES
        is  confirmed.
        And OXYGEN_LINE is assigned to OXYGEN_LINE

RULE :  Rule INVESTIGATE_GALLIUM_ABS (#58)
If
        there is evidence of GALLIUM_PRESENT
        And SPECTRA.TYPE is "ABS"
Then GALLIUM_ANALYZED_ABS
        is  confirmed.
        And CODE is set to "GA"
        And Execute
"FINDAREA"(@ATOMID=SPECTRA.POINTS,GALLIUM_CALIBRATION_PEAK
.WAVENUMBER,CODE.Value;)
        And (GALLIUM_CALIBRATION_PEAK.AREA*2.780e+14)
                is assigned to GALLIUM_CALIBRATION_PEAK.CONC
```

```
RULE : Rule INVESTIGATE_GALLIUM_PHOTO (#59)
If
        there is evidence of GALLIUM_PRESENT
        And SPECTRA.TYPE is "PHOTO"
Then GALLIUM_ANALYZED_PHOTO
        is confirmed.
        And CODE is set to "GA"
        And 0.0 is assigned to GALLIUM_CALIBRATION_PEAK.AREA

RULE : Rule GALLIUM_LINE_X4 (#60)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than 422
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                424
Then GALLIUM_LINE
        is confirmed.
        And GA_LINE_X4.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_X4.WN

RULE : Rule GALLIUM_LINE_X2 (#61)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than 380
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                382
Then GALLIUM_LINE
        is confirmed.
        And GA_LINE_X2.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_X2.WN
```

```
RULE :  Rule GALLIUM_LINE_X1 (#62)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  346
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                348
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_X1.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_X1.WN

RULE :  Rule GALLIUM_LINE_EX (#63)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  459
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                461
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_EX.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_EX.WN

RULE :  Rule GALLIUM_LINE_ESTAR (#64)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  941
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                942
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_ESTAR.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_ESTAR.WN
```

```
RULE :  Rule GALLIUM_LINE_EI (#65)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  586
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                587.2
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_EI.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_EI.WN

RULE :  Rule GALLIUM_LINE_9 (#66)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  576
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                578.2
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_9.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_9.WN

RULE :  Rule GALLIUM_LINE_8 (#67)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  573
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                576
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_8.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_8.WN
```

```
RULE :  Rule GALLIUM_LINE_7 (#68)
I f
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  570.5
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                571.5
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_7.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_7.WN

RULE :  Rule GALLIUM_LINE_6 (#69)
I f
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  567
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                569.5
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_6.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_6.WN

RULE :  Rule GALLIUM_LINE_5P (#70)
I f
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  933.5
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                934.5
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_5P.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_5P.WN
```

```
RULE :  Rule GALLIUM_LINE_5 (#71)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  563.7
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                564.7
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_5.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_5.WN

RULE :  Rule GALLIUM_LINE_4P (#72)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  929.5
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                930.5
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_4P.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_4P.WN

RULE :  Rule GALLIUM_LINE_4 (#73)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  547.7
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                553
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_4.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_4.WN
```

```
RULE :  Rule GALLIUM_LINE_3P (#74)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  921
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                922
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_3P.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_3P.WN

RULE :  Rule GALLIUM_LINE_3 (#75)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  541
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                542
Then GALLIUM_LINE
        is  confirmed.
        And GA_L!NE_3.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_3.WN

RULE :  Rule GALLIUM_LINE_2P (#76)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  896.7
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                897.7
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_2P.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_2P.WN
```

```
RULE :  Rule GALLIUM_LINE_11 (#77)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  584
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                586.3
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_11.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_11.WN

RULE :  Rule GALLIUM_LINE_10 (#78)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  579
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                584
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_10.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_10.WN

RULE :  Rule GALLIUM_LINE_1 (#79)
If
        there is evidence of GALLIUM_PRESENT
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than  469.25
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                470.25
Then GALLIUM_LINE
        is  confirmed.
        And GA_LINE_1.EXIST is set to TRUE
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GA_LINE_1.WN
```

```
RULE : Rule INVESTIGATE_GALLIUM (#80)
I f
        there is evidence of POSSIBLE_GALLIUM
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is greater
                than 468.8
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is less than
                470.8
        And <<|POSSIBLE_GALLIUM_PEAKS|>>.WAVENUMBER is greater
                than 540.5
        And <<|POSSIBLE_GALLIUM_PEAKS|>>.WAVENUMBER is less than
                542.5
        And <<<|POSSIBLE_GALLIUM_PEAKS|>>>.WAVENUMBER is greater
                than 547.2
        And <<<|POSSIBLE_GALLIUM_PEAKS|>>>.WAVENUMBER is less
                than 549.2
Then GALLIUM_PRESENT
        is confirmed.
        And <|POSSIBLE_GALLIUM_PEAKS|>.WAVENUMBER is assigned to
                GALLIUM_CALIBRATION_PEAK.WAVENUMBER

RULE : Rule INVESTIGATE_INDIUM_ABS (#81)
I f
        there is evidence of INDIUM_PRESENT
        And SPECTRA.TYPE is "ABS"
Then INDIUM_ANALYZED_ABS
        is confirmed.
        And CODE is set to "IN"
        And Execute
"FINDAREA"(@ATOMID=SPECTRA.POINTS,INDIUM_CALIBRATION_PEAK.
WAVENUMBER,CODE.Value;)
        And (INDIUM_CALIBRATION_PEAK.AREA*2.0000000000e+15) is
                assigned to INDIUM_CALIBRATION_PEAK.CONC

RULE : Rule INVESTIGATE_INDIUM_PHOTO (#82)
I f
        there is evidence of INDIUM_PRESENT
        And SPECTRA.TYPE is "PHOTO"
Then INDIUM_ANALYZED_PHOTO
        is confirmed.
        And CODE is set to "IN"
        And 0.0 is assigned to INDIUM_CALIBRATION_PEAK.AREA
```

```
RULE : Rule INDIUM_LINE_X4 (#83)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                872
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than 874
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_X4.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_X4.WN

RULE : Rule INDIUM_LINE_X3 (#84)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                862
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than 864
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_X3.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_X3.WN

RULE : Rule INDIUM_LINE_X2 (#85)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                830
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than 832
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_X2.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_X2.WN
```

```
RULE :  Rule INDIUM_LINE_X1 (#86)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                798
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than 780
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_X1.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_X1.WN

RULE :  Rule INDIUM_LINE_EX (#87)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                908
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than 910
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_EX.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_EX.WN

RULE :  Rule INDIUM_LINE_ESTAR (#88)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1609.2
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1610.2
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_ESTAR.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_ESTAR.WN
```

```
RULE :  Rule INDIUM_LINE_EI (#89)
I f
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1254.3
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1255.3
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_EI.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_EI.WN

RULE :  Rule INDIUM_LINE_9 (#90)
I f
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1245.3
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1246.3
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_9.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_9.WN

RULE :  Rule INDIUM_LINE_8 (#91)
I f
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1241
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1244
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_8.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_8.WN
```

```
RULE : Rule INDIUM_LINE_7 (#92)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1238.5
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1239.5
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_7.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_7.WN

RULE : Rule INDIUM_LINE_6 (#93)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1236
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1237
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_6.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_6.WN

RULE : Rule INDIUM_LINE_5 (#94)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1231.7
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1232.7
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_5.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_5.WN
```

```
RULE :  Rule INDIUM_LINE_4P (#95)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1598
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1599
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_4P.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_4P.WN

RULE :  Rule INDIUM_LINE_4A (#96)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1217.5
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1218.5
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_4A.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_4A.WN

RULE :  Rule INDIUM_LINE_4 (#97)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1215.3
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1217.3
Then INDIUM_LINE
        is confirmed.
        And IN_LINE_4.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_4.WN
```

```
RULE :  Rule INDIUM_LINE_3P (#98)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1589.3
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1590.3
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_3P.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_3P.WN

RULE :  Rule INDIUM_LINE_3 (#99)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1207.5
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1208.5
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_3.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_3.WN

RULE :  Rule INDIUM_LINE_2P (#100)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1565
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1567
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_2P.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_2P.WN
```

```
RULE :  Rule INDIUM_LINE_2 (#101)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1175.3
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1176.3
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_2.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_2.WN

RULE :  Rule INDIUM_LINE_11 (#102)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1253
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1255
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_11.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_11.WN

RULE :  Rule INDIUM_LINE_10 (#103)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1247.4
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1251.4
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_10.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_10.WN
```

```
RULE :  Rule INDIUM_LINE_1 (#104)
If
        there is evidence of INDIUM_PRESENT
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1145
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1146
Then INDIUM_LINE
        is  confirmed.
        And IN_LINE_1.EXIST is set to TRUE
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is assigned to
                IN_LINE_1.WN

RULE :  Rule INVESTIGATE_INDIUM (#105)
If
        there is evidence of POSSIBLE_INDIUM
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is greater than
                1144.4
        And <|POSSIBLE_INDIUM_PEAKS|>.WAVENUMBER is less than
                1146.4
        And <<|POSSIBLE_INDIUM_PEAKS|>>.WAVENUMBER is greater
                than  1174.8
        And <<|POSSIBLE_INDIUM_PEAKS|>>.WAVENUMBER is less than
                1176.8
        And <<<|POSSIBLE_INDIUM_PEAKS|>>>.WAVENUMBER is greater
                than  1215.3
        And <<<|POSSIBLE_INDIUM_PEAKS|>>>.WAVENUMBER is less than
                1217.3
Then INDIUM_PRESENT
        is  confirmed.
        And <<|POSSIBLE_INDIUM_PEAKS|>>.WAVENUMBER is assigned to
                INDIUM_CALIBRATION_PEAK.WAVENUMBER
```

```
RULE : Rule INVESTIGATE_OXYGEN_ABS (#106)
If
        there is evidence of OXYGEN_PRESENT
        And SPECTRA.TYPE is "ABS"
Then OXYGEN_ANALYZED_ABS
        is confirmed.
        And CODE is set to "OX"
        And Execute
"FINDAREA"(@ATOMID=SPECTRA.POINTS,OXYGEN_CALIBRATION_PEAK.
WAVENUMBER,CODE.Value;)
        And (OXYGEN_CALIBRATION_PEAK.AREA*3.090e+16) is assigned
                to OXYGEN_CALIBRATION_PEAK.CONC

RULE : Rule INVESTIGATE_OXYGEN_PHOTO (#107)
If
        there is evidence of OXYGEN_PRESENT
        And SPECTRA.TYPE is "PHOTO"
Then OXYGEN_ANALYZED_PHOTO
        is confirmed.
        And CODE is set to "OX"
        And 0.0 is assigned to OXYGEN_CALIBRATION_PEAK.AREA

RULE : Rule OXYGEN_LINE_1 (#108)
If
        there is evidence of OXYGEN_PRESENT
        And <|POSSIBLE_OXYGEN_PEAKS|>.WAVENUMBER is greater than
                1135.0
        And <|POSSIBLE_OXYGEN_PEAKS|>.WAVENUMBER is less than
                1137.0
Then OXYGEN_LINE
        is confirmed.
        And OX_LINE_1.EXIST is set to TRUE
        And <|POSSIBLE_OXYGEN_PEAKS|>.WAVENUMBER is assigned to
                OX_LINE_1.WN
```

```
RULE : Rule INVESTIGATE_OXYGEN (#109)
If
        there is evidence of POSSIBLE_OXYGEN
        And <|POSSIBLE_OXYGEN_PEAKS|>.WAVENUMBER is greater than
                1135.0
        And <|POSSIBLE_OXYGEN_PEAKS|>.WAVENUMBER is less than
                1137.0
Then OXYGEN_PRESENT
        is confirmed.
        And <|POSSIBLE_OXYGEN_PEAKS|>.WAVENUMBER is assigned to
                OXYGEN_CALIBRATION_PEAK.WAVENUMBER

RULE : Rule EXAMINE_THE_POSSIBLE_PEAKS (#110)
If
        GALLIUM_ANALYZED_ABS is not NOTKNOWN
        And GALLIUM_ANALYZED_PHOTO is not NOTKNOWN
        And BORON_ANALYZED_ABS is not NOTKNOWN
        And BORON_ANALYZED_PHOTO is not NOTKNOWN
        And INDIUM_ANALYZED_ABS is not NOTKNOWN
        And INDIUM_ANALYZED_PHOTO is not NOTKNOWN
        And ALUMINUM_ANALYZED_ABS is not NOTKNOWN
        And ALUMINUM_ANALYZED_PHOTO is not NOTKNOWN
        And CARBON_ANALYZED_ABS is not NOTKNOWN
        And CARBON_ANALYZED_PHOTO is not NOTKNOWN
        And OXYGEN_ANALYZED_ABS is not NOTKNOWN
        And OXYGEN_ANALYZED_PHOTO is not NOTKNOWN
Then PEAKS_ANALYZED
        is confirmed.

RULE :  Rule Force_the_impurity_analysis (#111)
If
        POSSIBLE_ALUMINUM is not NOTKNOWN
        And POSSIBLE_GALLIUM is not NOTKNOWN
        And POSSIBLE_INDIUM is not NOTKNOWN
        And POSSIBLE_BORON is not NOTKNOWN
        And POSSIBLE_CARBON is not NOTKNOWN
        And POSSIBLE_OXYGEN is not NOTKNOWN
        And POSSIBLE_UNKNOWN_PEAKS is not NOTKNOWN
Then PEAKS_EXAMINED
        is confirmed.
```

```
RULE :  Rule Load_the_peak_table (#112)
I f
        there is evidence of SPECTRA_ID_ASKED
        And SPECTRA.TYPE is "ABS","PHOTO"
Then PEAKS_LOADED
        is  confirmed.
        And Retrieve from "@V(SPECTRA.PEAKS)"
@TYPE=NXPDB;@FILL=ADD;@UNKNOWN=TRUE;@NAME="!NAME!";
@CREATE=|POSSIBLE_PEAKS|;@PROPS=AMPLITUDE,WAVENUMBER,
HALF_WIDTH,AREA;@FIELDS="AMPLITUDE","WAVENUMBER",
"HALF_WIDTH","AREA";

RULE :  Rule PICK_POSSIBLE_ALUMINUM_PEAKS (#113)
I f
        <|POSSIBLE_PEAKS|>.WAVENUMBER is greater than 337
        And <|POSSIBLE_PEAKS|>.WAVENUMBER is less than 912
Then POSSIBLE_ALUMINUM
        is  confirmed.
        And Create Object <|POSSIBLE_PEAKS|>
        |POSSIBLE_ALUMINUM_PEAKS|

RULE :  Rule PICK_OUT_POSSIBLE_BORON_PEAKS (#114)
I f
        <|POSSIBLE_PEAKS|>.WAVENUMBER is greater than 183
        And <|POSSIBLE_PEAKS|>.WAVENUMBER is less than 713.5
Then POSSIBLE_BORON
        is  confirmed.
        And Create Object <|POSSIBLE_PEAKS|>
        |POSSIBLE_BORON_PEAKS|

RULE :  Rule PICK_POSSIBLE_CARBON_PEAK (#115)
I f
        <|POSSIBLE_PEAKS|>.WAVENUMBER is greater than 599.0
        And <|POSSIBLE_PEAKS|>.WAVENUMBER is less than 601.0
Then POSSIBLE_CARBON
        is  confirmed.
        And Create Object <|POSSIBLE_PEAKS|>
        |POSSIBLE_CARBON_PEAKS|
RULE :  Rule P!CK_POSSIBLE_GALLIUM_PEAKS (#116)
If
        <|POSSIBLE_PEAKS|>.WAVENUMBER is greater than 346
        And <|POSSIBLE_PEAKS|>.WAVENUMBER is less than 942
Then POSSIBLE_GALLIUM
        is  confirmed.
        And Create Object <|POSSIBLE_PEAKS|>
        |POSSIBLE_GALLIUM_PEAKS|

RULE :  Rule PICK_OUT_POSSIBLE_INDIUM_PEAKS (#117)
If
        <|POSSIBLE_PEAKS|>.WAVENUMBER is greater than 798
        And <|POSSIBLE_PEAKS|>.WAVENUMBER is less than 1610
Then POSSIBLE_INDIUM
        is  confirmed.
        And Create Object <|POSSIBLE_PEAKS|>
        |POSSIBLE_INDIUM_PEAKS|

RULE :  Rule PICK_POSSIBLE_OXYGEN_PEAKS (#118)
If
        <|POSSIBLE_PEAKS|>.WAVENUMBER is greater than 1135.0
        And <|POSSIBLE_PEAKS|>.WAVENUMBER is less than 1137.0
Then POSSIBLE_OXYGEN
        is  confirmed.
        And Create Object <|POSSIBLE_PEAKS|>
        |POSSIBLE_OXYGEN_PEAKS|
RULE :  Rule UNIDENTIFIED_PEAKS (#119)
If
        <|POSSIBLE_PEAKS|> is not a member of
                <|POSSIBLE_ALUMINUM_PEAKS|>
        And <|POSSIBLE_PEAKS|> is not a member of
                <|POSSIBLE_BORON_PEAKS|>
        And <|POSSIBLE_PEAKS|> is not a member of
                <|POSSIBLE_INDIUM_PEAKS|>
        And <|POSSIBLE_PEAKS|> is not a member of
                <|POSSIBLE_GALLIUM_PEAKS|>
        And <|POSSIBLE_PEAKS|> is not a member of
                <|POSSIBLE_CARBON_PEAKS|>
        And <|POSSIBLE_PEAKS|> is not a member of
                <|POSSIBLE_OXYGEN_PEAKS|>
Then POSSIBLE_UNKNOWN_PEAKS
        is  confirmed.
        And Create Object <|POSSIBLE_PEAKS|> |UNIDENTIFIED_PEAKS|

RULE :  Rule PLOT_THE_SPECTRA (#120)
If
        REPORT_ALUMINUM is not NOTKNOWN
        And REPORT_BORON is not NOTKNOWN
        And REPORT_GALLIUM is not NOTKNOWN
        And REPORT_INDIUM is not NOTKNOWN
        And REPORT_OXYGEN is not NOTKNOWN
        And REPORT_CARBON is not NOTKNOWN
Then  reports_done
        is  confirmed.
        And Execute "rgraph"(@ATOMID=SPECTRA.NAME;)

RULE :   Rule Get_the_spectra_ID (#121)
If
        SPECTRA.NAME is assigned to SPECTRA.NAME
Then SPECTRA_ID_ASKED
        is  confirmed.
        And Execute "FILLSPECTRA"(@ATOMID=SPECTRA.NAME;)
        And Execute "PEAKPICK"()
        And STRCAT(SPECTRA.NAME,".MAS") is assigned to
                SPECTRA.POINTS
        And STRCAT(SPECTRA.NAME,".NXP") is assigned to
                SPECTRA.PEAKS
```

U.S. Government Printing Office 548-076